3D object model acquisition from silhouettes(Dissertation_全文)

AUTHOR(S):

Iiyama, Masaaki

3D Object Model Acquisition from Silhouettes

Masaaki Iiyama

March 2006

Abstract

This thesis proposes an approach for acquiring a 3D object model from silhouettes. The 3D object model we propose is a set of object's properties which are necessary for reproducing the object's appearance. The properties in the 3D object model are categorized into three properties; photometry modeled by reflection properties, geometry modeled by object's shape, and motion modeled by a sequential object's pose.

The 3D object model has two advantages, portability and suitability for preservation. These advantages push a widespread usage of the 3D object model. We categorize the usage into two types; applications which are putting importance on the portability, and applications which are putting importance on preserving objects' appearance. We can categorize the model acquisition from two types of view; computation time and observation time for model acquisition. The computation/observation time affect on a lighting environment we can use; whether lighting environment is known or unknown, controllable or uncontrollable. There is a relationship between the computation/observation time and controllability of lighting environment.

The main contribution of this thesis is to make the relationship clear through the following three research topics.

We first propose a method to acquire the shape and reflection properties under unknown and varying lighting environment. The method has flexibility for parallels processing which enables real-time processing. Previously proposed methods had acquired reflection properties by using an object's shape. Our method does not use the object's shape and acquires both the shape and the reflection properties. Our method acquires the object's shape from silhouettes, using the volume intersection method. The shape is acquired as a set of voxels and named a visual hull. Our method also acquires the reflection properties of each voxel.

Previously proposed methods, which acquired both the shape and the

reflection properties, require heavy computational cost. In our method, the calculation is done independently at each voxel; it reduces the computational cost. The shape is acquired by voxel-independent calculation in our method. And the reflection properties are also acquired by voxel-independent calculation, using our reflection model which is suitable for voxel-independent calculation.

Secondly we propose a method for acquiring the motion of articulate objects from visual hulls. Several methods to estimate an articulated motion had been proposed. These methods require a shape model of each body parts which compose an articulated object. Our method does not use any shape model and measures both the shape of body parts and the articulated motion at the same time.

Visual hulls acquired in time sequences are used to acquire the shape and the motion. All the voxel included in a body part are always under the same rigid motion. We extract such voxels from the whole shape. Making a correspondence between voxels acquired in different times provides us the extraction, but unnecessary voxels in the visual hull makes the correspondence difficult. Our solution for this difficulty is the use of multi-dimensional distance of each voxel. The multi-dimensional distance contains distances from the voxel to the visual hull's surface, and the distances are calculated along several directions. The distances along some directions will receive effects from the unnecessary voxels. The distances along the other directions receive no effect from them. The use of some part of multi-dimensional distance instead of using whole multi-dimensional distance overcomes the effect from unnecessary voxels.

Finally, we propose a method which reconstructs smooth or concave surface, which the volume intersection method can not reconstruct. The visual hull, which is acquired by the volume intersection method, is a convex hull circumscribing the object. Acquiring a concave surface with the volume intersection method is impossible. The volume intersection method requires many cameras in order to acquire a smooth surface, even if the surface is not a concave surface.

We employ photometric stereo to acquire the smooth and concave surfaces. Photometric stereo estimates surface normals as a needle map by controlling lighting environment. The needle map contains surface normals of concave and smooth surfaces. In case when the needle maps contain depth edges, incorrect depth maps are reconstructed. An incorrect depth map is not consistent with silhouettes which are taken from other viewpoints. Based

on this fact, our method minimizes two types of energy function to reconstruct the depth map: One energy function is based on a consistency between depth map and needle map, and the other is based on a consistency between depth map and silhouettes.

.

Acknowledgements

This work was carried out at Graduate School of Informatics, Kyoto University during the years 2000-2006.

I would like to express my deepest gratitude to Professor Michihiko Minoh for giving me the opportunity to carry out this thesis. He gave me many constructive suggestions and encouragements during the whole study. I am also very grateful to my thesis committee-members, Professor Takashi Matsuyama and Professor Yuichi Nakamura, for their valuable comments. I would like to thank Associate Professor Koh Kakusho and Associate Professor Yoshinari Kameda at University of Tsukuba for providing constructive comments and suggestions.

I wish to thank current and former members in Minoh laboratory: Keisuke Yagi, Dr. Satoshi Nishiguchi, Dr. Masayuki Murakami, Dr. Tetsuo Shoji, Yoko Yamakata, Takuya Funatomi and Masahiro Toyoura for helping me with brilliant ideas and comments. Thanks are also due to all the members of Minoh laboratory for their helpful discussions on my research.

Finally, I would like to thank my wife and daughter, Satoko and Haruka, for their patience, help and understanding. Without their help I would not finish this thesis.

Contents

1 Introduction **1**
1.1 3D model acquisition under unknown lighting environment . . 4
1.2 Shape and motion acquisition without controling lighting environments 5
1.3 Shape acquisition with controlable lighting environments . . . 6

2 3D Model Acquisiton under Unknown Lighting Environment **9**
2.1 Introduction 9
2.2 3D Shape Reconstruction 11
2.2.1 Target Space and Camera Model 11
2.2.2 Visual Hull 11
2.2.3 Voxel-independence of Volume Intersection Method 14
2.3 Reflection Model for Voxel-independent Calculation 15
2.3.1 Torrance-Sparrow Model 15
2.3.2 Simplified Torrance-Sparrow Model 15
2.4 Voxel-independent Reconstruction 18
2.4.1 Extracting Surface Voxels 18
2.4.2 Normal Vector Estimation 19
2.4.3 Color Parameter Estimation 21
2.5 Experiments and Discussion 23
2.5.1 Experiments 23
2.5.2 Discussion 29
2.6 Conclusion 30

3 Shape and Motion Acquisition without Controling Lighting Environments **31**
3.1 Introduction 31

3.2 Shape of Body Parts and Pose of Articulate Objects 33
3.3 Voxel Feature for Region Correspondence 34
3.3.1 $d(V_{t_i}, x, \boldsymbol{q}_k)$ 36
3.3.2 Dissimilarity 37
3.4 EM Argorithm 39
3.4.1 E-step 41
3.4.2 M-step 43
3.4.3 Initial Estimate of Probability 44
3.4.4 Segmentation 45
3.5 Experiment 45
3.5.1 Experiment with Synthesis Data 45
3.5.2 Experiment with Real Data 48
3.6 Conclusion 48

4 Shape Acquisition with Controlable Lighting Environments 57
4.1 Introduction 57
4.2 Photometric Stereo 59
4.2.1 Distance Map Reconstruction from Needle Map 61
4.3 Shape Reconstruction by Using Silhouette and Needle Map Consistency 62
4.3.1 Needle Map Consistency 64
4.3.2 Silhouette Consistency 66
4.3.3 Minimization 70
4.3.4 Unifying the Distance Maps 70
4.4 Experiments 71
4.4.1 Experiments with Synthetic Data 71
4.4.2 Experiments with Real Data 77
4.5 Conclusion 81

5 Conclusion 85

Chapter 1

Introduction

This thesis presents an approach for acquiring a 3D object model from silhouettes. The 3D object model we propose is a set of object's properties which are necessary for reproducing the object's appearance.

The object's appearance varies with various factors: Objects vary their appearance, varying their *pose* at all times. Lighting environments also vary the object's appearance. Object's surface reflects incident lights, *Reflection properties* provide the object's appearance. Viewpoints also vary the object's appearance. When the viewpoint is given, *object's shape* provides the appearance.

These factors are modeled by the properties included in the 3D object model. The properties in 3D object model are categorized into three properties; photometry modeled by reflection properties, geometry modeled by object's shape, and motion modeled by a sequential object's pose. A 3D object model with these three properties enables us creating a view of the object under arbitrary lighting condition, view point, and object's pose.

The 3D object model has the following advantages.

- Portability: The 3D object model can be transmitted to remote places through a network.
- Suitability for Preservation: The 3D object model does not degrade, keeping the appearance of the original object.

These advantages push a widespread usage of the 3D object model. Applications of the 3D object model include teleconferences, virtual museums, and rapid prototyping.

These applications are categorized into the following two types. The first type is applications which put more importance on the portability rather than accuracy. Teleconference systems and distant learning systems are examples of it. These systems observe a scene by cameras, and transmit an appearance of the scene to distant places. A real-time processing is required for the transmission.

The second type is applications which put utmost importance on preserving objects' appearance. The digital museums and the rapid prototyping, which require realistic appearances, are major applications of this type. They do not require any real-time processing, only require accuracy.

We can categorize the model acquisition from two types of view; computation time and observation time for model acquisition. 3D object model acquisition is categolized into three types of acquistions.

- Acquiring 3D object models of moving objects with real-time processing(e.g. teleconference systems). It is sensitive to computation and observation time.

- Acquiring 3D object models of moving objects with offline processing (e.g. virtual museums). It is sensitive to observation time and not sensitive to computation time.

- Acquiring 3D object models of static (not moving) objects with offline processing (e.g. rapid prototyping). It is not sensitive to computition or observation time.

We propose three methods to cope with these different types of problems.

The computation/observation time affect on a lighting environment we can use. In the real-time processing system, for example, object models should be acquired under a time-varying lighting environment. It is difficult to control the lighting environment in real-time system, because of the object's motion during the control. In the offline processing system for acquiring a model of moving object, it is also difficult to control the lighting environment, but not difficult to estimate the lighting environment. In the offline processing system for acquiring a model of static object, it is not difficult to control the lighting environment. The control of the lighting environment will improve the accuracy of the 3D object model. These examples imply that there is a relationship between the computation/observation time and controllability of lighting environment. When we put more importance on

the computation time than the controlling lighting conditions, object model will be less accurate. On the other hand, when we put more importance on the accuracy and control lighting conditions, real-time processing is hard to accomplish.

The main contribution of this thesis is to make the relationship clear through the following three research topics.

- Real-time 3D model acquisition under unknown and varying lighting environment.
- Shape and motion acquisition without controlling lighting environment.
- Shape acquisition with controllable lighting environments.

We define and acquire the 3D object model with the following properties. As for the reflection properties, diffuse and specular reflection properties, which are widely used properties, are included in the 3D object model. And as for the motion, we focus on articulate objects and acquire the articulate motion. The articulate motion is a set of rigid motion of each body parts of the object.

Versatile approach, in other words, an approach which can acquire 3D models of various objects, is also required for 3D object model acquisition. The use of silhouettes for acquiring 3D object models satisfies the requirement, for object's silhouettes can be robustly extracted. The silhouettes also have an advantage; they can be extracted robustly under various lighting envirionments.

Acquisition of 3D object models from silhouettes is hard task, however. Restrictions as to an object's shape or prepared object's shape are required. Many works have acquired object's reflectance properties by using a pre-measured object's shape[46, 45, 37]. Many works had acquired object's motion by using a rough shape of an object[44, 12, 11, 6, 33, 40]. Shape from silhouettes has a restriction on object's surfaces[20, 21, 22]; it can not measure any concave surface.

The second contribution of this thesis is to remove the restriction and preparation. In chapter 2, we show an approach by which the reflectance properties and the shape are acquired at the same time. The approach does not require any pre-measured shape or pre-measured lighting environment. In addition, our method employs a new reflectance model and reduces the computational cost by voxel-independent calculation; it is necessary for real-time

computation. An approach which acquires the object's articulated motion without using prepared object's shape is described in chapter 3. Non-rigid motion makes it difficult to acquire the articulated motion. Our method employs a probabilistic approach to fix the non-rigid motion. A method for measuring the concave and smooth surface is described in chapter 4. We use photometric-stereo to acquire the smooth/concave surfaces by controlling the lighting environment. The use of needle map acquired with photometric stereo and silhouettes.

1.1 3D model acquisition under unknown lighting environment

Previously proposed methods had acquired reflection properties by using an object's shape. Our method does not use the object's shape and acquires both the shape and the reflection properties.

Our method acquires the object's shape from silhouettes, using the volume intersection method[26, 28]. The shape is acquired as a set of voxels and named a visual hull. Our method also acquires the reflection properties of each voxel.

Previously proposed methods, which acquired both the shape and the reflection properties, require heavy computational cost. In our method, the calculation is done independently at each voxel; it reduces the computational cost. The shape is acquired by voxel-independent calculation in our method. And the reflection properties are also acquired by voxel-independent calculation, using our reflection model which is suitable for voxel-independent calculation.

Reflection properties, including a diffuse reflection and a specular reflection [39], of each *surface voxel* are reconstructed. The surface voxels is a voxel located on the surface of the visual hull. Based on the Torrance-Sparrow reflection model[31], we propose an improved reflection model which is suitable for the voxel-independent reconstruction. Parameters of our reflection model are acquired under unknown lighting condition. Our method estimates directions of every light source at each frame; even if the lighting environment varies, our method will still work.

1.2 Shape and motion acquisition without controling lighting environments

Our method acquires the motion of an articulated object. Several methods to estimate an articulated motion had been proposed[44, 12, 11, 6, 33, 40]. These methods require a shape model of each body parts which compose an articulated object. The requirement is not suitable for the 3D object model acquisition, because the applications described in section 1 require 3D models of *various* objects and preparing the shape model of the objects takes a lot of costs. Our method does not use any shape model and measures both the shape of body parts and the articulated motion at the same time.

In our method, a whole shape of an articulated object is acquired as a visual hull which consists of a set of voxels. Visual hulls acquired in time sequences are used to acquire the shape and the motion. All the voxel included in a body part are always under the same rigid motion. We extract such voxels from the whole shape. Making a correspondence between voxels acquired in different times provides us the extraction, but unnecessary voxels in the visual hull makes the correspondence difficult.

Our solution for this difficulty is the use of multi-dimensional distance of each voxel. The multi-dimensional distance contains distances from the voxel to the visual hull's surface, and the distances are calculated along several directions. The distances along some directions will receive effects from the unnecessary voxels. The distances along the other directions receive no effect from them. The use of some part of multi-dimensional distance instead of using whole multi-dimensional distance overcomes the effect from unnecessary voxels.

Non-rigid motion observed around joints of an articulate object also has bad effect on the motion estimation. Each body part of articulate objects has rigid motion. Non-rigid region, which is the region under non-rigid motion, is also included in the articulated objects, however. Previously proposed methods did not take in account the non-rigid motion, regarding that the articulate objects only have rigid parts. When their methods try to estimate the motion of the articulate objects including non-rigid motion, their estimations have bad accuracy.

To solve the problem, we employ a probabilistic approach. Our approach does not directly estimate the shape of each body part. Instead of this, it estimates a probability that each voxel belongs to the body parts. The use

of voxels weighted by the probability gives more accurate rigid motion.

1.3 Shape acquisition with controlable lighting environments

The volume intersection method has an advantage over the other methods. The advantage is that the method can acquire shapes of texture-less objects. The volume intersection method requires object's silhouettes and does not require point correspondence, which other methods require. Extracting silhouettes of texture-less object is easier task than obtaining point correspondence from texture-less objects.

The volume intersection method has also a disadvantage. The disadvantage is the difficulty of acquiring smooth and concave surfaces[20, 21, 22]. The visual hull, which is acquired by the volume intersection method, is a convex hull circumscribing the object. Acquiring a concave surface with the volume intersection method is impossible. The volume intersection method requires many cameras in order to acquire a smooth surface, even if the surface is not a concave surface.

We employ photometric stereo[35, 3] to acquire the smooth and concave surfaces. Photometric stereo estimates surface normals as a needle map by controling lighting environment, and the needle map contains surface normals of concave and smooth surfaces.

The needle map acquired by photometric stereo does not directly express a shape of the object, however. Reconstruction of a distance map from the needle map is required. Maximizing the following consistency gives the distance map. The consistency is that the needle map is consistent with the surface normal derived from the reconstructed distance map. We call the consistency needle map consistency.

Depth edges[34] make it difficult to calculate the consistency. A depth edge is an area on the distance map; on the area, a depth from camera to the surface varies discontinuously. The discontinuity disables the calculation of surface normal; it means that existence of depth edges disables the calculation of the needle map consistency.

We propose an approach which uses silhouettes taken from different viewpoints. The silhouettes reduce the bad effects of depth edge. An incorrect depth map produced by depth edge is not consistent with the silhouettes

which are taken from other viewpoints. Based on this fact, our method minimizes two types of energy functions to reconstruct the depth image: one energy function is based on a consistency between a depth image and a needle map, and the other is based on a consistency between a depth image and silhouettes.

Chapter 2

3D Model Acquisiton under Unknown Lighting Environment

In this chapter, we propose an approach to reconstruct a 3D object shape and reflection property with real-time processing under unknown lighting condition. Previously proposed methods require heavy computational cost. In our approach, the calculation of reconstruction is done independently at each voxel, and computational cost is reduced. The volume intersection method reconstructs the shape by voxel-independent calculation. Based on Torrance-Sparrow reflection model[31], we propose an improved reflection model which is suitable for the voxel-independent reconstruction. Parameter of out reflection model is estimated by voxel-independent calculation. Reconstruction process consists of three steps: First, the surface voxels are extracted. Then surface normal at each surface voxel is calculated. Finally, its reflection property is estimated.

2.1 Introduction

Improvement in processing power of computers enables us to copy real scene into virtual 3D shape in computers[16]. Our goal is to reconstruct the whole scene automatically. Once the scene is reconstructed, everyone outside the real space can observe it from any viewpoint. The reconstructed scene should be photorealistic and the reconstruction should be processed in real-time to

follow scene changes.

We consider the situation in which multiple cameras surround the real scene. Recently, several approaches which can reconstruct photorealistic scene from multiple images have already been proposed. They are classified into image-based approaches [19, 38, 23, 29, 24, 1] and model-based approaches [46, 45, 37].

The image-based approaches reproduce virtual images directly from pixels of images. For example, Levoy and Hanrahan and Gortler *et al.* developed a realistic reconstruction[23, 10].However, these methods employ a dense set of images, so they are not suitable for real-time reconstruction of a real scene. Seitz and Dyer proposed a different approach[38] that assigns a color to each voxel, namely a spatial point, though their approach can only be applied to objects which has only diffuse surface, and therefore specular reflection on the object surface e.g. highlight cannot be recovered.

On the other hand, model-based approaches employed shape models and surface reflection models. With these models, the reflectance properties at points on a surface of objects are estimated. Sato et al.[37] developed an object shape and the reflectance modeling technique, which can successfully estimate the reflectance property on the object surface with a dense set of images taken in simple lighting conditions. The lighting conditions are composed of only a single measured light source, though the lighting condition of the real scene is composed of many light sources which are hard to measure. Thereby their approach cannot be applied to the real space reconstruction. Yu et al.[46, 45] proposed a real scene reconstruction method under unknown complex lighting conditions with a not so dense set of images of the scene. However, it is not applicable to the real-time reconstruction due to huge amount of computation.

We propose a new method for reconstruction of a real space with multiple cameras. The characteristics of our approach are:

- Voxel-based reconstruction with multiple cameras based on the volume intersection method. It has flexibility for parallels processing which enables high-speed shape calculation.

- Color reconstruction including not only diffuse reflectance property but also specular reflectance property under unknown lighting condition. It needs no a priori modeling of objects and lights in the real space, and enables more photorealistic rendering than that only with the diffuse reflectance property.

In order to realize high-speed reconstruction, color reconstruction is performed independently at each voxel in the same way as the shape reconstruction. A simplifies reflection model based on the dichromatic reflection model is introduced for this purpose.

2.2 3D Shape Reconstruction

2.2.1 Target Space and Camera Model

Our method reconstructs the shape of objects in a *target space* (Figure.2.1). Appropriate target space is determined by cameras' position and cameras' view angle. The target space consists of a set of voxels, all of which have same size. Let us write the voxel by $\mathcal{V}$, let the center of $\mathcal{V}$ to be $\boldsymbol{v}$, the number of cameras to be n, and each camera to be $C_1, \ldots, C_n$. We use cameras which follow pinhole camera model. The position and direction of each camera are calibrated as a 3×4 perspective projection matrix P_i beforehand. A projection matrix P_i projects a 3D point $\boldsymbol{M}$ to 2D point $\boldsymbol{p_i}$ on the camera C_i's image.

$$\boldsymbol{p_i} = P_i \boldsymbol{M} \tag{2.1}$$

2.2.2 Visual Hull

The volume intersection method requires silhouettes observed from various viewpoints. Let us denote the silhouette taken in camera C_i by $\mathfrak{R}_i$. The silhouette is extracted by calculating color differences between the input image and the background image taken in advance.

Let us consider the case when an object is imaged by a camera C_i and its silhouette is extracted as $\mathfrak{R}_i$. The object is circumscribed by a cone; a cone whose apex is at the viewpoint of camera C_i and whose silhouette is $\mathfrak{R}_i$. We call the cone the *visual cone* for camera C_i and denote it as $VC_{\boldsymbol{C_i}}$.

In the case when the object is observed by m cameras, it exists within the product of all VC_{C_i} (Figure.2.1, Figure.2.2). We denote it by ESS and name it a visual hull. The ESS consists of a set of voxels, and it is written as,

$$ESS = \{\boldsymbol{v} \mid \boldsymbol{p} = \boldsymbol{P}_i \boldsymbol{v},\ \boldsymbol{p} \in \mathfrak{R}_i\} \tag{2.2}$$

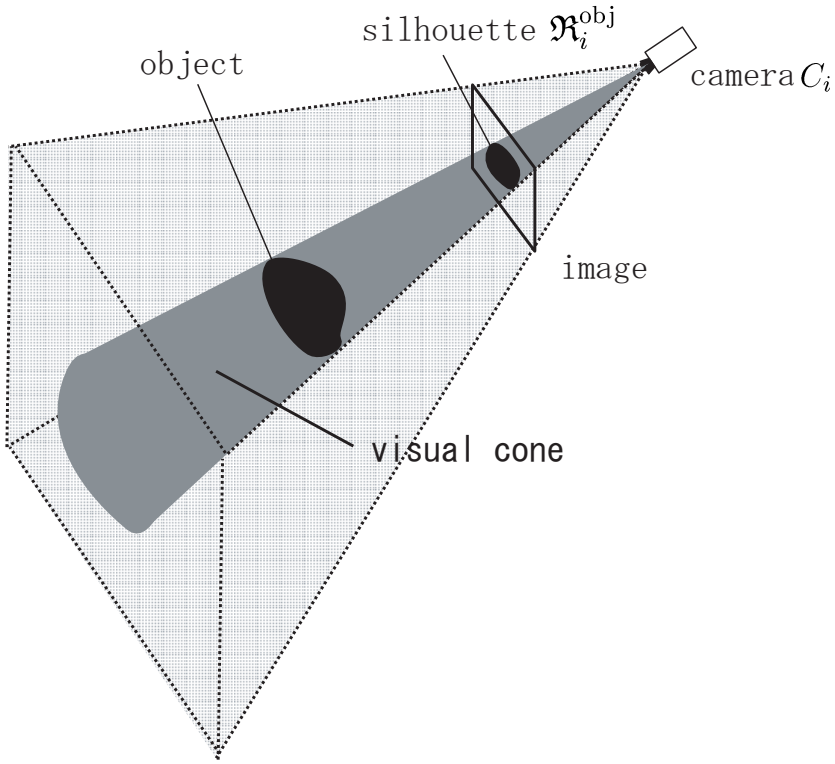


Figure 2.1: Visual Cone and Visual Hull

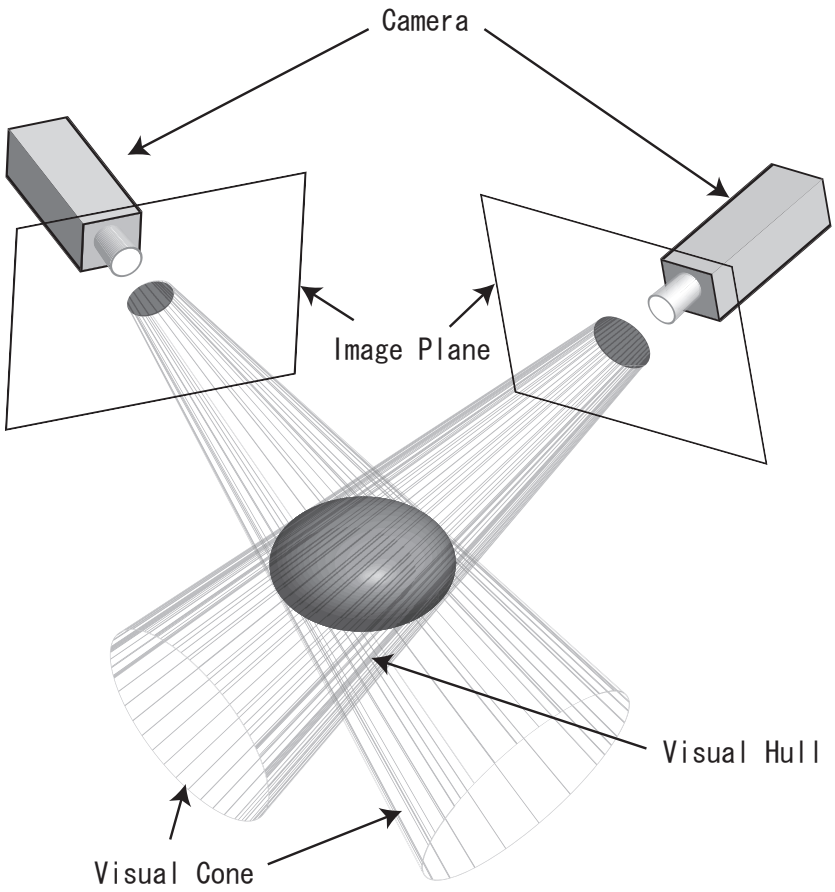


Figure 2.2: Visual Hull and Object Shape

Generally speaking, surfaces of the visual hull do not match with object's surfaces. They only circumscribe the object's surfaces. However, the use of a large enough number of cameras lets the visual hull approximate the object's shape.

Some kind of surfaces can not be reconstructed by the volume intersection method[18, 20, 21, 22]. Its necessary and sufficient condition had been given by Lauentini [20].

Proposition 1 *A necessary and sufficient condition for a point M belonging to the surface of a visual hull to belong to the object's surface is that at least one line L passes through M without intersecting the visual hull at any other point.*

No concave surface can be reconstructed, even if huge amount of cameras are used. Reconstruction of smooth surface requires a lot of cameras.

2.2.3 Voxel-independence of Volume Intersection Method

The target space is divided into the voxels. The determination whether each voxel is included in a visual hull provides the visual hull. The equation

$$\boldsymbol{P}_i\boldsymbol{v} \in \mathfrak{R}_i, \ \forall\, i = 1, \ldots, n \tag{2.3}$$

gives the determination.

The equation requires no voxels except v, it means that the determination could be done voxel-independently. That is, the volume intersection method is a voxel-independent method.

We should notice that the volume intersection method has voxel-independency but does not have pixel-independency. More than two voxels refer the same pixel, and it loses the pixel-independency.

The pixel-independency can not be a large problem, however. The reason is that the amount of image data is very small as compared with the amount of voxel data. Copying the image data as many as the number of processors enables a parallel computation and requires little computational costs[15].

2.3 Reflection Model for Voxel-independent Calculation

A light reflected on an object's surface is recognized as the surface's color. The reflection is often modeled by the dicromatic reflection model[39]. The dicromatic reflection model describes the reflected light with two types of reflection; a specular reflection and a diffuse reflection. Several models based on the dicromatic reflection model have been proposed, but they are not suitable for voxel-independent calculation. We propose a new model which is suitable for voxel-independent calculation, simplifying the Torrance-Sparrow model[31]. The Torrance-Sparrow model, which is based on the dicromatic reflection model, has a good approximation of real objects' surfaces.

2.3.1 Torrance-Sparrow Model

We denote a light strength by 3D vector $\boldsymbol{I} = (I_R, I_G, I_B)^\top$. Let strengths of point lights $L_1, \ldots, L_m$ to be $\boldsymbol{I}^{\text{in}}{}_j = ({I^{\text{in}}}_{jR}, {I^{\text{in}}}_{jG}, {I^{\text{in}}}_{jB})$. Strengthes of reflection of L_j, we denote it by $\boldsymbol{I}^{\text{ref}} = ({I^{\text{ref}}}_R, {I^{\text{ref}}}_G, {I^{\text{ref}}}_B)$, is

$$
\begin{aligned}
I_c^{\text{ref}} &= \sum_{j=1}^{m} I_{jc}^{\text{in}} k_c^{\text{diff}} \cos\theta(\boldsymbol{n}, \boldsymbol{l}_j) \\
&\quad + \sum_{j=1}^{m} I_{jc}^{\text{in}} * k_c^{\text{spec}} \frac{1}{\cos\psi(\boldsymbol{n}, \boldsymbol{e})} e^{-\frac{\phi(\boldsymbol{n}, \boldsymbol{l}_j, \boldsymbol{e})^2}{2\sigma^2}} \\
&\qquad\qquad (c = R, G, B)
\end{aligned} \tag{2.4}
$$

,where, $\boldsymbol{n}$ is a surface normal, $\boldsymbol{e}$ is an observing direction, $\boldsymbol{l}_j$ is a lighting direction of L_j, $\theta(\boldsymbol{n}, \boldsymbol{l}_j)$ is an angle between $\boldsymbol{n}$ and $\boldsymbol{l}_j$, $\psi(\boldsymbol{n}, \boldsymbol{e})$ is an angle between $\boldsymbol{n}$ and $\boldsymbol{e}$, $\phi(\boldsymbol{n}, \boldsymbol{l}_j, \boldsymbol{e})$ is an angle between $\boldsymbol{n}$ and a bisector of $\boldsymbol{l}_j$ and $\boldsymbol{e}$. ${k^{\text{diff}}}_c$ and ${k^{\text{spec}}}_c$ are a diffuse reflection coefficient and a specular reflection coefficient respectively, and σ is directivity parameter of specular reflection. Figure2.3 illustrates the reflection.

2.3.2 Simplified Torrance-Sparrow Model

Equation2.4 contains two terms; a diffuse reflection term and a specular reflection term. The equation shows the following properties.

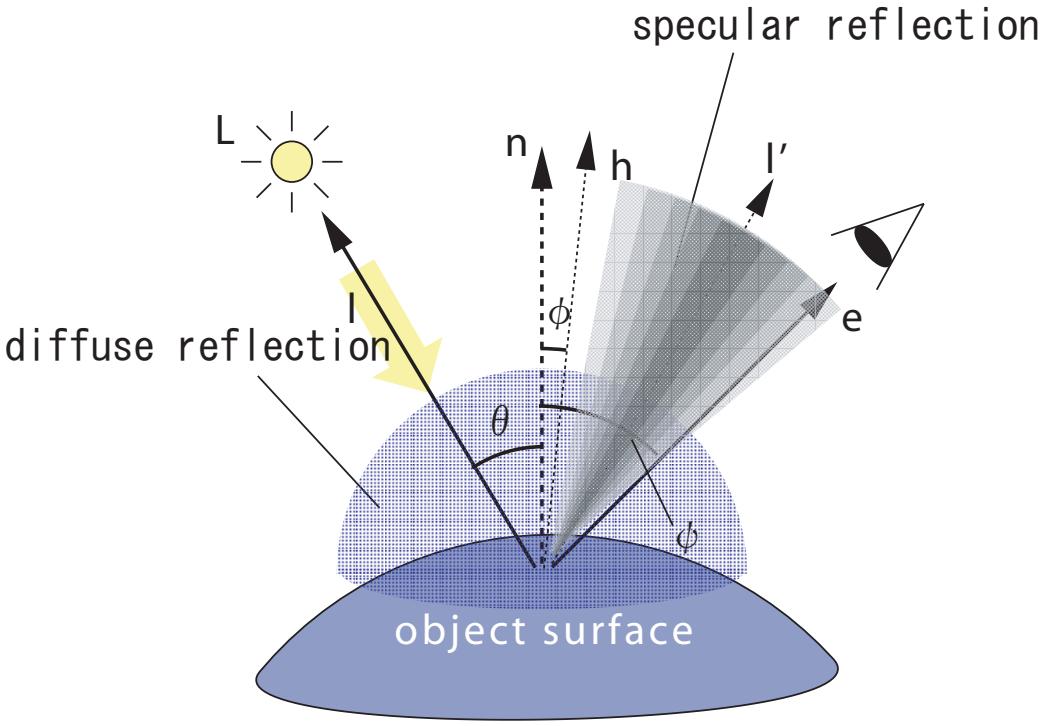


Figure 2.3: Light Reflection.

(i) Diffuse reflections are uniform in the observing directions. Every viewpoint observes the reflection as same strength.

(ii) Specular reflections have highly directivity, which is controlled by σ, and are observed only from narrow observing directions. A viewpoint which has an observing direction which exists around a mirror direction of an incident light can observe the specular reflection.

Based on these properties and Equation2.4, we formulate a new reflection model which is suitable for the voxel-independent calculation.

The first right term of Equation2.4, representing diffuse reflection, does not contain observing direction $\boldsymbol{e}$. We name the term **diffuse color** and denote it by $\boldsymbol{I}^{\mathrm{diff}}$.

The second property shows the following fact; when the lights are not densely arranged and a viewpoint observes a specular reflection, only one light gives the specular reflection.

Let us suppose that a specular reflection on a surface is observed, and let L_s gives the specular reflection. For all the light L_j except for L_s, $\phi(\boldsymbol{n}, \boldsymbol{l}_j, \boldsymbol{e})$ will become $\phi(\boldsymbol{n}, \boldsymbol{l}_j, \boldsymbol{e}) \gg 0$, and $e^{-\frac{\phi(\boldsymbol{n},\boldsymbol{l}_j,\boldsymbol{e})^2}{2\sigma^2}}$ will be approximately 0. It approximates the second right term of Equation2.4 which represents specular reflection by

$$I_{sc}^{\mathrm{in}} k_c^{\mathrm{spec}} \frac{1}{\cos\psi(\boldsymbol{n}, \boldsymbol{e})} e^{-\frac{\phi(\boldsymbol{n},\boldsymbol{l}_s,\boldsymbol{e})^2}{2\sigma^2}} \qquad (c = R, G, B). \tag{2.5}$$

Our model expresses the strength of incident light I_{sc}^{in} and the specular reflection coefficient k_c^{spec} by single parameter I_c^{spec}.

$$I_c^{\mathrm{spec}} = I_{sc}^{\mathrm{in}} k_c^{\mathrm{spec}} \quad (c = R, G, B)$$

We name $\boldsymbol{I}^{\mathrm{spec}}$ **specular color**.

The diffuse color and the specular color rewrite Equation2.4.

$$\boldsymbol{I}^{\mathrm{ref}} = \boldsymbol{I}^{\mathrm{diff}} + E(\boldsymbol{n}, \boldsymbol{l}_s, \boldsymbol{e}) \boldsymbol{I}^{\mathrm{spec}} \tag{2.6}$$

where,

$$E(\boldsymbol{n}, \boldsymbol{l}_s, \boldsymbol{e}) = \frac{1}{\cos\psi(\boldsymbol{n}, \boldsymbol{e})} e^{-\frac{\phi(\boldsymbol{n},\boldsymbol{l}_s,\boldsymbol{e})^2}{2\sigma^2}} \tag{2.7}$$

The observing direction $\boldsymbol{e}$ is given by a viewpoint, and the surface normal $\boldsymbol{n}$ is given by acquired shape. We also estimate remaining three parameters; a direction of incident light L_s denoted by $\boldsymbol{l}_s$, the diffuse color $\boldsymbol{I}^{\mathrm{diff}}$ and the specular color $\boldsymbol{I}^{\mathrm{spec}}$. We name the diffuse and specular color the **color parameters**. Every surface voxel has its own color parameters, and it implies that an estimation of color parameters can be done voxel-independently.

2.4 Voxel-independent Reconstruction

The shape of object is reconstructed as a set of voxel by using the volume intersection method described in section2.2. We express the surface of the object by a set of voxels located on the surface of visual hull. We call these voxels *surface voxels*. Every surface voxel has its own refection properties, and the properties are expressed by the color parameters described in section2.3.

In this section, our method for reconstructing the shape and the color parameters is described.

The reconstruction procedure consists of three steps.

In the first step, the surface voxels are extracted. In the second step, normal of each surface voxel is estimated. The normal of surface voxels is necessary for estimating the color parameters. We write the normal of voxel $\mathcal{V}$ by $\boldsymbol{n}(\mathcal{V})$. The color parameters, a diffuse color and a specular color, and a direction of an incident light are estimated in the last step.

2.4.1 Extracting Surface Voxels

A voxel-independent calculation extracts the surface voxels.

Two conditions described in the following give the determination whether a voxel $\mathcal{V}$ is a surface voxel or not. When we project a surface voxel into each camera image, its projected region satisfies the following condition.

(i) For all cameras i, the projected region is included in its silhouette $\mathfrak{R}_i$. It means that every surface voxel is included in the visual hull.

(ii) For at least one camera i, the projected region is on the edge of its silhouette $\mathfrak{R}_i$. It means that every surface voxel is on the surface of the visual hull.

Here, we define the edge of $\mathfrak{R}_i$ as a set of pixels which are included in $\mathfrak{R}_i$ and some of whose 4-neighbor pixels are not included in $\mathfrak{R}_i$.

All surface voxels satisfy both conditions. We call a voxel which does not statisfy (i) an **empty voxel**, and call a voxel which only satisfies (ii) an **internal voxel**.

The determination of the surface voxels can be done by voxel-independent calculation, because the two conditions only use the projection matrix $\boldsymbol{P}_i$ and the silhouette $\mathfrak{R}_i$.

2.4.2 Normal Vector Estimation

The visual hull is the product of all visual cones for all cameras, and surfaces of the visual cones compose visual hull's surface. The surface voxels are on the surface of visual hull; they are also on the surface on a visual cone. We can easily determine a visual cone on which a surface voxel is located: The second condition described in previous section shows that a surface voxel is projected on the edge of at least one of the silhouettes. When a surface voxel is projected on the edge of $\mathfrak{R}_i$, it is located on the surface of camera C_i's visual cone. A surface normal of the visual cone gives a normal of the surface voxel.

A surface normal of a surface voxel is acquired by voxel-independent calculation. When we project the surface voxel into the camera images, it is projected on the edge of at least one of the silhouettes. We call a pixel on the edge an *edge pixel*. An edge pixel has its 2D surface normal, and the normal can be extracted only from the silhouette; so the normal can be extracted voxel-independently. The surface normal of the surface voxel is parallel to the 2D surface normal, and it is orthogonal to a view line on the edge pixel (Figure2.4). The edge pixel and its 2D normal give the surface normal. We write the surface normal by $\boldsymbol{n}_i(\mathcal{V})$.

Some surface voxels are projected on the edge of multiple silhouettes; each silhouette may give different surface normals. In such case, we use averaged surface normal as the surface normal.

The surface normal described above can be acquired only from the silhouettes and the projection matrices, and any other voxels are not necessary; it is acquired by voxel-independent calculation.

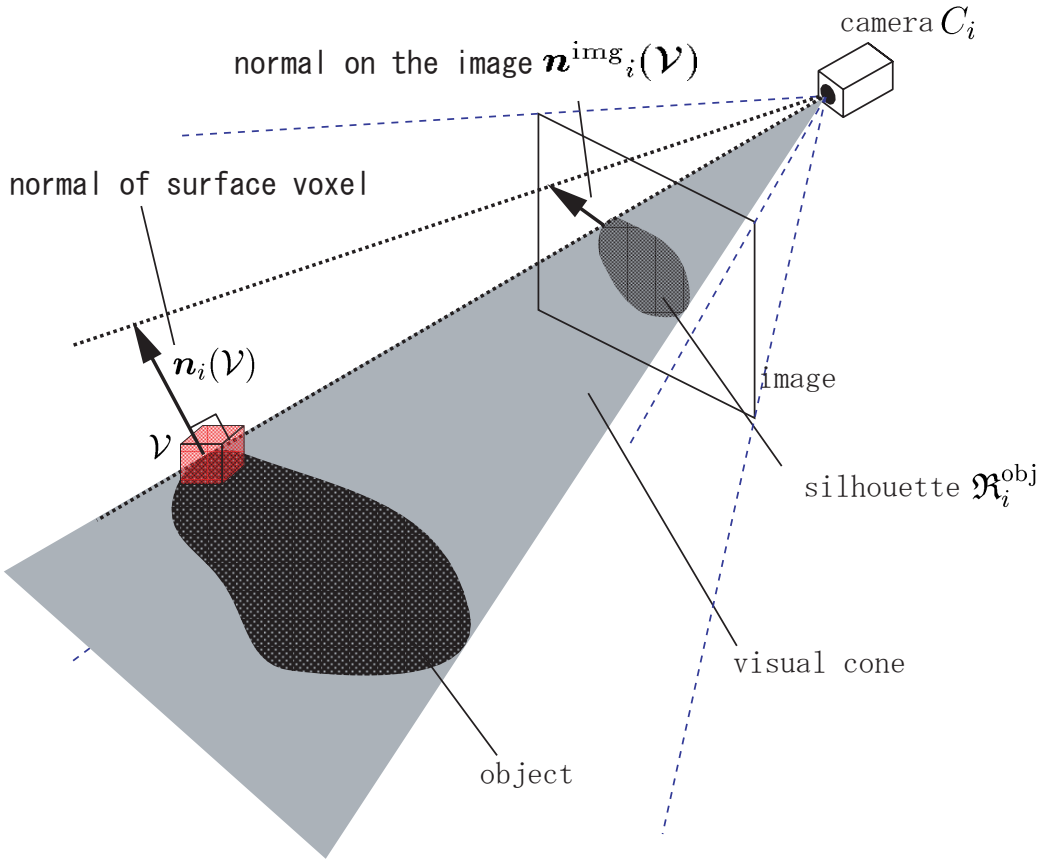


Figure 2.4: A planar normal on image plane and a spatial normal on the surface of a visual hull.

2.4.3 Color Parameter Estimation

Each surface voxel $\mathcal{V}$ has the color parameters, $\boldsymbol{I}^{\text{diff}}$ and $\boldsymbol{I}^{\text{spec}}$, as written in Equation2.6. The color parameters are given by cameras which observe the $\mathcal{V}$.

We first determine the cameras which observe $\mathcal{V}$, and then estimate the color parameters from these cameras.

Observing Cameras Extraction

Not all the camera observes $\mathcal{V}$, because other voxels may occlude $\mathcal{V}$. We call a camera which observes a surface voxel $\mathcal{V}$ an **observing camera** of $\mathcal{V}$. The observing cameras are extracted in the following procedure: First, a surface normal of $\mathcal{V}$ is extracted. Second, a plane whose normal is equal to the surface normal of $\mathcal{V}$ and which passes through $\mathcal{V}$ is acquired. Finally, cameras facing in front of the plane are extracted as the observing cameras.

We denote the number of observing cameras by n^{obs}, and denote the observing cameras by $C_{i_k}(1 \leq i_k \leq n,\ 1 \leq k \leq n^{\text{obs}})$.

$\mathcal{V}$ is projected into a pixel of each observing camera's silhouette. When camera C_{i_k} observes $\mathcal{V}$, the color of a pixel on the camera C_{i_k} represents the light reflected on $\mathcal{V}$. We call the color an **observed color** by camera C_{i_k} and denote it as $\boldsymbol{I}_{i_k}$.

Color Parameter Estimation from Observed Color

Our reflection model described in Equation2.6 contains two colors; the diffuse color and the specular color. It requires the separation of observed color $\boldsymbol{I}_{i_k}$ into two colors.

To solve the separation, we focus on the directivity of specular reflection, described in Section2.3.2, and a coefficient E in $\boldsymbol{I}^{\text{spec}}$.

Equation2.7 shows that, when observing direction $\boldsymbol{e}_j$ has large difference with a mirror direction of an incident light $\boldsymbol{l}'$, $\phi(\boldsymbol{n}, \boldsymbol{l}_s, \boldsymbol{e}_i)$ becomes to be $\phi(\boldsymbol{n}, \boldsymbol{l}_s, \boldsymbol{e}_i) \gg 0$ and E becomes to be approximately 0. In other words, when an observing camera C_{i_k} satisfies $\boldsymbol{e}_{i_k} \simeq \boldsymbol{l}'_s$ and cameras and lights are not densely arranged, every E of the other observing cameras becomes to be approximately 0.

It implies that at most one camera observes a specular reflection on $\mathcal{V}$ from light $\boldsymbol{l}_s$. We call such camera a **specular observing camera** of $\mathcal{V}$.

Determining the specular observing camera separates the observed color into the two colors.

Determining the Specular Observing Camera

Compared between the diffuse color and the specular color, specular color has highly bright color. When n^{obs} is larger than 2, at least one camera observes only the diffuse color and does not observe the specular color. And in such case, at most one camera observes highly bright color and the other cameras observe approximately same color. Comparing the brightness of each observed colors, we determine the camera which observes the highly bright color as the specular observing camera.

The procedure is described in the following. First, we calculate the difference between an observed color and averaged observed color. The difference is expressed as a 3D color vector $\boldsymbol{d}_{i_k}$.

$$\boldsymbol{d}_{i_k} = \boldsymbol{I}_{i_k} - \frac{1}{n^{\text{obs}} - 1} \sum_{j \neq k} \boldsymbol{I}_{i_j} \tag{2.8}$$

Second, average of the three elements of $\boldsymbol{d}_{i_k}$, we denote it as $b(\boldsymbol{I}_{i_k})$, is acquired. Third, we find a observing camera which has the largest $b(\boldsymbol{I}_{i_k})$. We call the camera specular observing camera candidate and denote it as $C_{i_{\text{spec}}}$. Finally, a specular observing camera candidate whose $b(\boldsymbol{I}_{i_k})$is larger than a given threshold δ^{spec} is extracted as a specular observing camera.

When no specular observing camera candidate is extracted as a specular observing camera, we determine the $\mathcal{V}$ has only the diffuse color. We call such voxel a **single color voxel**, and the diffuse color of a single color voxel is given by

$$\boldsymbol{I}^{\text{diff}} = \frac{1}{n^{\text{obs}}} \sum_{k=1}^{n^{\text{obs}}} \boldsymbol{I}_{i_k} \tag{2.9}$$

$$\boldsymbol{I}^{\text{spec}} = \boldsymbol{0}. \tag{2.10}$$

On the other hand, a voxel which has both specular color and diffuse color is called a **multiple color voxel**.

Color Parameters of Multiple Color Voxel

Only one camera $C_{i_{\text{spec}}}$ observes the specular color from light $\boldsymbol{l}_s$. The lighting direction $\boldsymbol{l}_s$ and a mirror direction of an observing direction are similar.

Approximating the $\boldsymbol{l}_s$ by the mirror direction of an observing direction, we acquire an estimate of $\boldsymbol{l}_s$,

$$\tilde{\boldsymbol{l}_s} = 2(\boldsymbol{n} \cdot \boldsymbol{e}_{i_{\mathrm{spec}}})\boldsymbol{n} - \boldsymbol{e}_{i_{\mathrm{spec}}} \tag{2.11}$$

where, lengths of $\boldsymbol{n}$, $\boldsymbol{e}_{i_{\mathrm{spec}}}$, and $\tilde{\boldsymbol{l}_s}$ are normalized to be 1.

Observing cameras except for specular observing camera do not contain the specular color. The diffuse color $\boldsymbol{I}^{\mathrm{diff}}$ is given by

$$\boldsymbol{I}^{\mathrm{diff}} = \frac{1}{n^{\mathrm{obs}} - 1} \sum_{i_k \neq i_{\mathrm{spec}}} \boldsymbol{I}_{i_k} \tag{2.12}$$

Approximation of the $\boldsymbol{l}_s$ with Equation2.11 makes $\phi(\boldsymbol{n}, \boldsymbol{l}_s, \boldsymbol{e}_{i_{\mathrm{spec}}})$ to be 0 (Figure2.3). It rewrites Equation2.7 by

$$E(\boldsymbol{l}_s, \boldsymbol{e}_{i_{\mathrm{spec}}}) = \frac{1}{\cos\psi(\boldsymbol{n}, \boldsymbol{e}_{i_{\mathrm{spec}}})} \tag{2.13}$$

Equation2.6 and Equation2.8 give the specular color $\boldsymbol{I}^{\mathrm{spec}}$ by

$$\begin{aligned} \boldsymbol{I}^{\mathrm{spec}} &= (\boldsymbol{I}_{i^{\mathrm{spec}}} - \boldsymbol{I}^{\mathrm{diff}}) \cos\psi(\boldsymbol{n}, \boldsymbol{e}_{i^{\mathrm{spec}}}) \\ &= \boldsymbol{d}_{i^{\mathrm{spec}}} \cos\psi(\boldsymbol{n}, \boldsymbol{e}_{i^{\mathrm{spec}}}) \end{aligned} \tag{2.14}$$

These estimations described above require no reference to the other voxel. It means that our method, which is described in this section, can be done by voxel-independent calculation.

2.5 Experiments and Discussion

In this section, we show the experimental results of our voxel-independent reconstruction method and discuss the accuracy and computational cost of the method.

2.5.1 Experiments

We set up eight SONY EVI video cameras in a lecture room. A camera layout in the room is shown in Figure2.5.

A polyvinyl chloride blue ball whose size is 22.5cm in diameter was used for the experiments. Each camera has 640×480 pixels and observes $60 \times 60 \times$

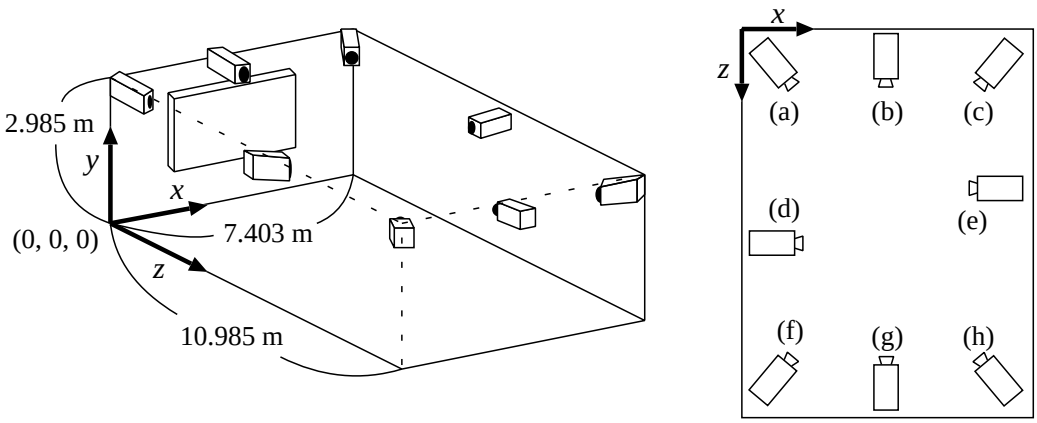
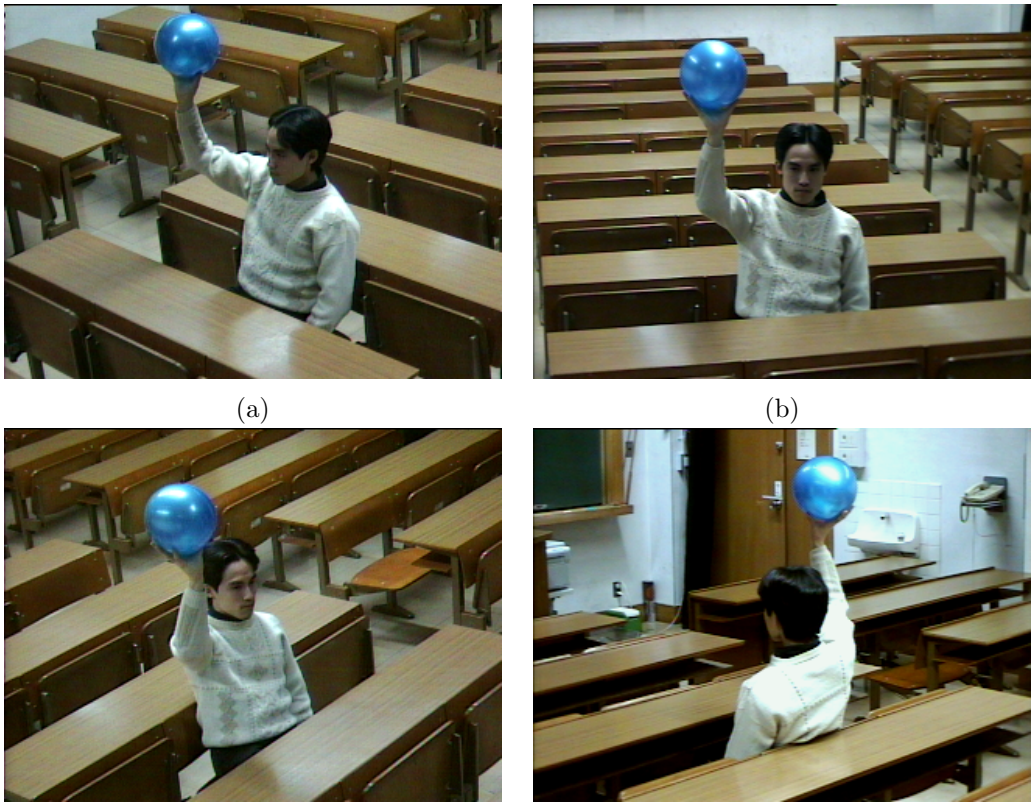


Figure 2.5: Camera layout.

60[cm] region. We set each voxel size as a cube of 0.5cm on a side. The total number of voxels is $120 \times 120 \times 120 = 1,728,000$.Each pixel on the images has RGB color value, whose range is $[0, 255]$. We set σ and δ^{spec} to be 0.5 and 50. Projection matrices of the cameras are calibrated by using Zhang's method[47].

The images taken by the eight cameras, which are illustrated in Figure2.5(a)-(h), are shown in Figure2.6. The shape and color property were reconstructed from them. The reconstruction results are shown in Figure2.7. The shape of the ball is reconstructed by using the visual hull method(Figure2.7(i)). Figure2.7(ii) and (iii) are synthesized views from the camera (e) and (f)'s viewpoint respectively. Figure2.7(iv) is a synthesized view, which synthesizes only the diffuse color, from the same camera's viewpoint as Figure2.7(ii). Synthesized views from viewpoints where the cameras are not arranged are shown in Figure2.7(v), (vi), (vii), and (viii).

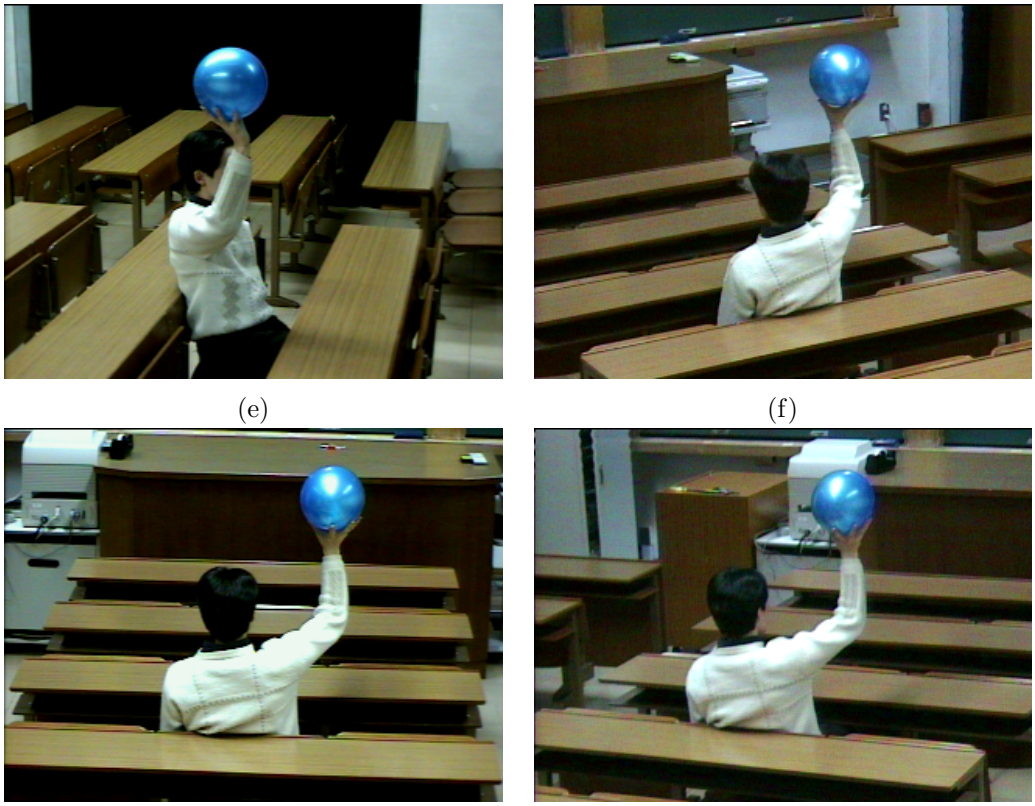
Table2.1 shows statistics of all processed voxels with respect to each voxel type, the empty voxel, the internal voxel, the single color voxel, and the multiple color voxel. Each processing time shown in the table is measured on a Sun Ultra2 Model2300 (UltraSPARC-II 300MHz CPU).



(a) (b)

(c) (d)

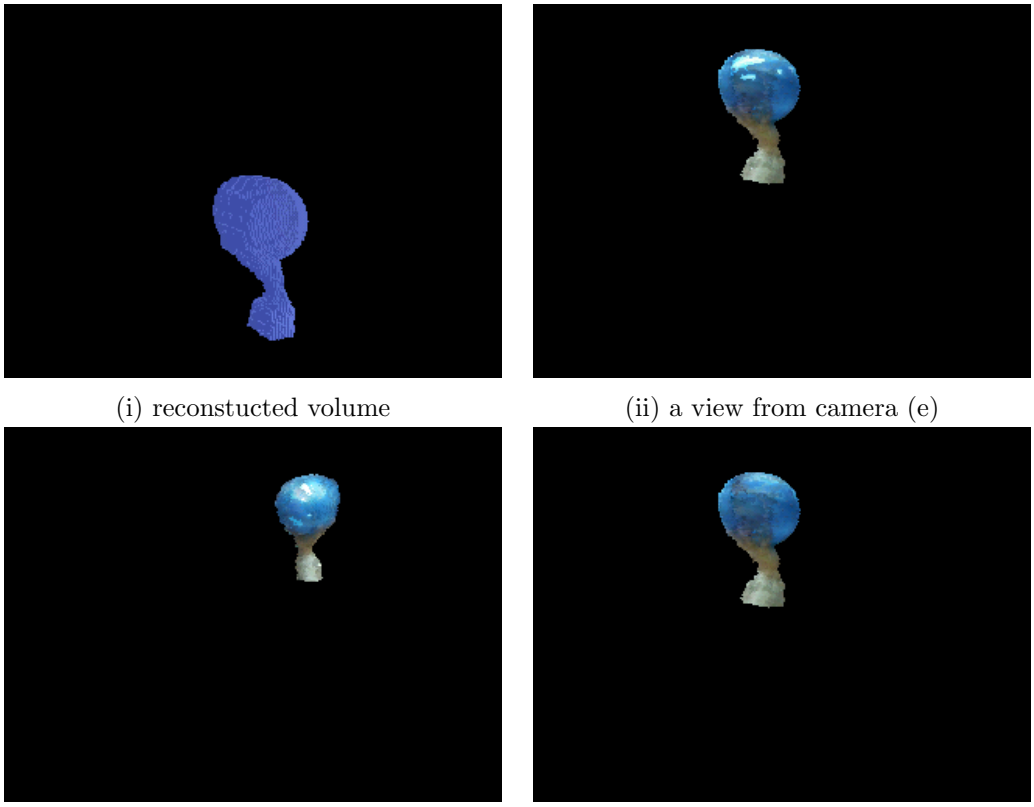
Figure 2.6: Input images



(e) (f)

(g) (h)

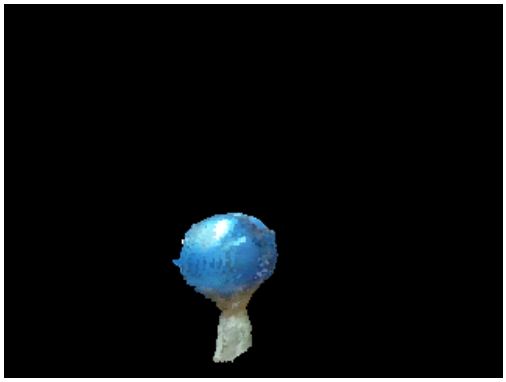
Figure 2.6: Input images



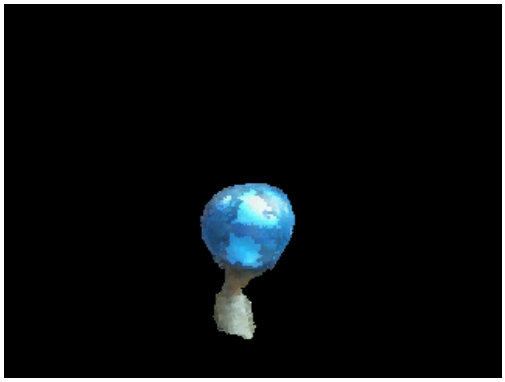
(i) reconstucted volume (ii) a view from camera (e)

(iii) a view from camera(f) (iv) diffuse color(camera(e))

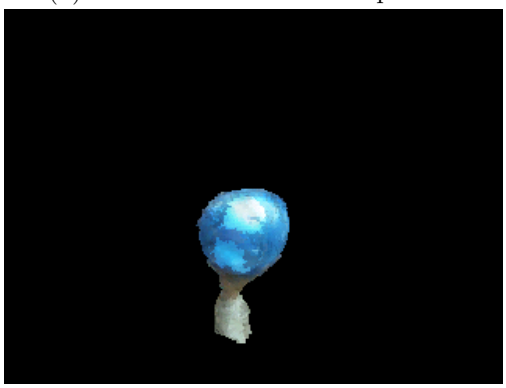
Figure 2.7: Reconstruction results.



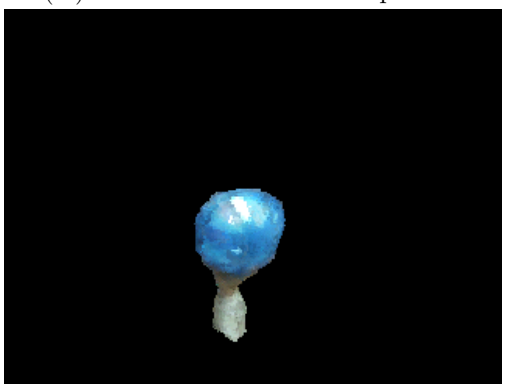
(v) a view from virtual viewpoint 1



(vi) a view from virtual viewpoint 2



(vii) a view from virtual viewpoint 3



(viii) a view from virtual viewpoint 4

Figure 2.7: Reconstruction results.

Table 2.1: Statistics of all processed voxels.

voxel type	the number of voxels	processing time [μsec]	total time [sec]
empty voxel	1,665,348	2.0486	3.4116
internal voxel	45,169	14.6772	0.6630
single color voxel ($n^{\text{obs}} = 1$)	26	68.2692	0.0018
single color voxel ($n^{\text{obs}} \geq 2$)	12,085	77.6035	0.9378
multiple color voxel ($n^{\text{obs}} \geq 2$)	5,372	86.4092	0.4642

2.5.2 Discussion

Comparison between Figure2.7 (ii) and (iv) shows that our method reconstructs not only a diffused color but also a specular color. As can be seen from the high-light on each image in Figure2.7, the position of the high-light is changed when the viewpoint is changed; this result also confirmed that our method reconstructs a specular color.

Figure2.7(vii) is a synthesized view from a viewpoint between a viewpoint of Figure2.7 (vi) and that of Figure2.7 (viii). Figure2.7(vii) has unnatural high-light, which spreads widely on the ball. The high-light is produced by a high-light on Figure2.7(vi) and that of Figure2.7(viii). The most likely explanation of the high-light is as follows: An incident light whose direction has small difference with a mirror direction of an observing direction produces a weak specular reflection. When such weak specular reflection is observed, our method estimates a direction of incident light to be the same as a mirror direction of an observing direction. Observing specular reflection of single incident light from two cameras leads to estimation failure; there are two incident lights. The failure poses the unnatural high-light.

One possible solution for avoiding the estimation failure is using a shading distribution on objects' surface. When a direction of incident light is the same as a mirror direction of an observing direction, the strongest specular reflection is observed. A voxel where the strongest specular reflection is observed has the brightest color among the neighbor voxels. Estimating a

direction of incident light by using the voxel where the brightest color is observed avoids the failure.

Our method does not estimate σ, which controls directivity of specular reflection. The use of a shading variation on objects' surface is one possible approach to estimate the σ.

High directivity of specular reflection increases the number of cameras which are necessary for observing specular reflection. On the contrary, when directivity is low, the presence of a large number of cameras causes multiple cameras to observe same specular light. The number of necessary cameras depends on a given parameter σ.

Table2.1 shows the following property as to processing time: A processing time for each empty voxel is short, but huge number of empty voxel results in spending much processing time. We will reduce the processing time by using an octree-based division of measurement space.

2.6 Conclusion

In this chapter, we proposed a method of 3D shape and color reconstruction of objects in the situation the multiple cameras surround a certain real scene. The input to our method is a set of images taken by surrounding cameras. These images reconstruct 3D shape and color property of objects in the real scene by pallalelized algorithm. The result is fine enough to reporduce highlight of objects derived from the specular reflection.

As we mentioned 2.2, the concave shape can not be reconstructed. Therefore reflection property of it can also not be reconstructed. Acquisition of them with voxel-independent approach is required as a future work.

Chapter 3

Shape and Motion Acquisition without Controling Lighting Environments

In this chapter, we discuss acquisition of the motion of articulate objects from visual hulls acquired in time sequences. Visual hulls contain unnecessary regions which are not a part of an object shape, and such regions have undesirable effects on motion acquisition. We use a voxel feature which reduces the undesirable effects. In addition, articulated objects have non-rigid regions which exist around their joints. The non-rigid region also has undesirable effects. Our probabilistic approach reduces the undesirable effects. We estimate a probability that a voxel belongs to a part of articulated objects. The use of voxels weighted by the probability for motion estimation solves the undesirable effects.

3.1 Introduction

Articulate objects consist of rigid body parts and non-rigid joints. Each body part is connected by the joints. The pose of the body part is described as a position and a direction of the body part, and a pose of articualted objects is defined as a set of body part's pose. The motion of articulate objects is modeled by a sequential object's pose.

Motion-based approaches for acquiring shape and pose of body parts have been proposed[43, 30, 17, 12, 7, 13, 14]. Our method is also based on the

approach. These approaches acquire the shape and pose under an assumption that the body parts are under different rigid motions. These approaches first acquire whole shape of the object in time sequence, and then extract a region which is under the same rigid motion. The extraction can be done by making a correspondence between the regions of the whole shapes acquired in different time. Extracted rigid region gives a shape of body part, and its rigid motion gives a pose of body part.

Our method requires whole shapes of the object in time sequence. Virtual museum, which is an application we anticipate, requires 3D model of *various* objects. A method which acquires the whole shapes of the various objects in time sequence is necessary for our method. The volume intersection method satisfies the needs, for it only requires silhouettes, which can be acquired robustly. Our method employs the volume intersection method to acquire the whole shapes of the object.

The visual hull is acquired as the whole shape and is expressed by a set of voxels. Making a correspondence between the voxels taken in different time gives the extraction of rigid region, and finally the shape and pose of body part are acquired.

The unnecessary voxels included in the visual hull has a bad effect on the correspondence. The visual hull, which is acquired as the whole shape, is a convex hull circumscribing the object, and it contains the unnecessary voxels which are not included in the object. The number of the unnnecessary voxels and their arrangement vary during the motion; it has a bad effect on the correspondence.

To avoid the bad effect, Cheung[17] and Gao[12] used color information. They extracted feature points on images by using color information, and made a correspondence between the feature points.

The use of color information spoils the advantage of the volume intersection method; the volume intersection method can acquire the shape of texture-less objects. The use of color information can not extract feature points of texture-less objects and can not also make their correspondence.

Our method extracts a shape feature without using color information and makes the correspondence, not spoiling the advantage of the volume intersection method. We make an assumption on the unnecessary voxels. The assumption is that only several parts of object's surface are occluded by the unnecessary voxels and the other parts are not occluded. Based on the assumption, we propose and use a multi-dimensional shape feature and its dissimilarity.

Our multi-dimensional shape feature is assigned to each voxels. It consists of a set of distance from the voxel to the visual hull's surface; each distance is calculated along different directions. When no unnecessary voxel exists, these distances do not change during the motion, for each body part is under the rigid motion. From the same reason, the multi-dimensional feature is also consistent during an object's rigid motion. However, when unnecessary voxels exist, they change some of the distance and make it difficult to make the correspondence. The dissimilarity of the feature solves the difficulty, using a part of the multi-dimensional distance instead of using whole multi-dimensional distance.

Non-rigid motion observed around the joints also has bad effect on the pose estimation. Non-rigid region, which is the region under non-rigid motion, is included in the articulated objects. Previously proposed methods did not take in account the non-rigid motion, regarding that the articulate objects only have rigid parts. When their methods try to estimate the motion of the articulate objects including non-rigid motion, they estimate the rigid motion of non-rigid motion and their estimation have bad accuracy.

To solve the problem, we employ a probabilistic approach. Our approach does not directly estimate the shape of each body part. Instead of this, it estimates a probability that each voxel belongs to the body parts. The use of voxels weighted by the probability gives more accurate rigid motion.

3.2 Shape of Body Parts and Pose of Articulate Objects

A whole shape of an articulate object is acquired with the volume intersection method described in Chapter2. We express 3D space as voxel space, and the visual hull as a set of voxels. Let us denote the whole shape of the articulate object at time $t_i (i = 0, \ldots, N)$ by V_{t_i}, and a voxel included in V_{t_0} by $x (x \in V_{t_0})$.

All the voxels in a body part are always under the same rigid motion. We extract such voxels from the whole shape V_{t_0}. A set of extracted voxels represents the region in which a body part exists at time t_0. In other words, it represents the shape of the body part.

Every voxel x included in V_{t_0} has a label, which makes a correspondence the voxel with one of the body parts. Let M be the number of body parts

and let each body parts be named as *part* $1, \ldots,$*part* M. A voxel labeled as m belongs to *part* m. Assigning a label to each voxel x gives the shape of each body parts at time t_0. In the following discussion, we denote the label which voxel x has by $z(x)$ and denote a set of labels for all the voxel included in V_{t_0} by $\boldsymbol{z}$.

$$\boldsymbol{z} = \{z(x) | x \in V_{t_0}\} \tag{3.1}$$

A body part's rigid motion from t_0 to t_i and its shape at t_0 give the region in which the body part exists at time t_i. Relative position between the position at t_0 and that at t_i is expressed as the rigid motion from t_0 to t_i; the rigid motion gives the pose of each body part at t_i. We name the position of each body part at t_0 *base position*, and express the rigid motion by $D_j^{t_i}$.

3.3 Voxel Feature for Region Correspondence

Making a correspondence between regions acquired in different time requires a feature which is extracted from the regions. It also requires that the feature should be stable during the motion.

Image based methods for making a correspondence between images have been proposed. They gave each pixel a feature value. The feature value was determined by a color value of the pixel. They had an assumption that the color value is stable during the motion.

We also give each voxel a feature value. The feature value is also required to be stable during the motion. Unnecessary voxels included in a visual hull makes it difficult to satisfy the requirement, because the number and arrangement of unnecessary voxels are unstable during the motion.

Another requirement to the feature exists. The requirement is that the feature has a single peak according to the position of x. The explanation of this is given by our searching method for voxel correspondence.

The searching method for voxel correspondence is described as follows: The feature values of two voxels give dissimilarity between the two voxels. Dissimilarity between a voxel and corresponding voxel is smaller than that between a voxel and non-corresponding voxel. Minimizing the dissimilarity gives the corresponding voxel. A dissimilarity which has multi-peak causes

the searching to get stuck in local minima. The use of a feature which has single peak avoids the multi-peak of dissimilarity.

Our method uses a feature which satisfies the two properties described above; being stable and single-peak. We focus on a distance from a center of voxel to a surface of visual hull. Here we define the surface of visual hull as visual hull border; voxels included in visual hull and voxels not included in the visual hull create the border.

A distance from a point which is inside of the object to a surface of the object is stable during the motion when the object is rigid. In the same manner, when the visual hull contains no unnecessary voxels, a distance from a center of voxel which is included in visual hull to a surface of visual hull is almost stable, in other words it contains small difference during the motion. A size of voxel affects the difference.

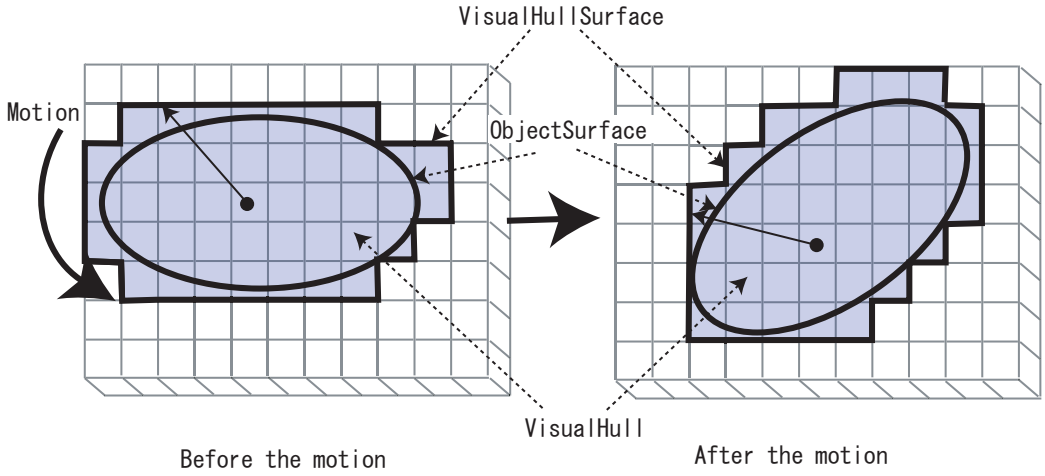


Figure 3.1: distance between object surface and internal point

However, the unnecessary voxels in the visual hull prevents the distance from satisfying the stablability, for the unnecessary voxels changes the distance. The visual hull contains unnecessary voxels; on the contrary some parts of the visual hull contain no unnecessary voxels. The use of such parts

keeps the stablability. It is difficult to know the parts, where no unnecessary voxels exist, however.

Our solution for this difficulty is the use of multi-dimensional distance. A multi-dimensional distance contains distances calculated along several directions. The distances along some directions get effects from the unnecessary voxels. The distances along other directions get no effect from it. The use of part of multi-dimensional distance instead of using whole multi-dimensional distance overcomes the effect from unnecessary voxels.

The multi-dimensional distance is calculated as follows: Let the number of directions and each direction to be s and $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_s$, where each $\boldsymbol{q}_k$ is normalized vector. The multi-dimensional distance contains distances calculated along $\boldsymbol{q}_k$ $(k = 1, \ldots, s)$.

The multi-dimensional distances of two voxels give their dissimilarity. Calculation of the dissimilarity omits some $\boldsymbol{q}_k$ to overcome the effect from unnecessary voxels. Let the number of omitting directions to be ρ. Even if ρ of s distances are not stable, the dissimilarity is still stable.

A voxel feature is defined in the following equation.

$$F_{V_{t_i}}(x; \boldsymbol{q}_1, \ldots, \boldsymbol{q}_s) = [d(V_{t_i}, x, \boldsymbol{q}_1), \ldots, d(V_{t_i}, x, \boldsymbol{q}_s)] \tag{3.2}$$

Where, $d(V_{t_i}, x, \boldsymbol{q}_k)$ described below is a function which is calculated from a distance between the center of voxel x and surface of the visual hull V_{t_i}.

3.3.1 $d(V_{t_i}, x, \boldsymbol{q}_k)$

A distance between the center of voxel x and the surface of the visual hull V_{t_i} gives $d(V_{t_i}, x, \boldsymbol{q}_k)$. The feature $F_{V_{t_i}}$ should have a single peak, which we mentioned before. $d(V_{t_i}, x, \boldsymbol{q}_k)$ should also have a single peak.

$d(V_{t_i}, x, \boldsymbol{q}_k)$ is calculated in the following procedure:

if x is included in V_{t_i} Suppose there are several line-segments which start at the center of x and end at the surface of V_{t_i}. The shortest line-segment of them gives the distance between the center of x and the surface of V_{t_i}. We define $d(V_{t_i}, x, \boldsymbol{q}_k)$ as the length of the shortest line-segment.

if x is not included in V_{t_i} In this case, some $\boldsymbol{q}_k$ may cause an absence of line-segment, which starts at the center of x and ends at the surface of

V_{t_i}. One example of it is shown on voxel B's $\boldsymbol{q}_1$ in Figure3.2. In order to fix the absence, we calculate the $d(V_{t_i}, x, -\boldsymbol{q}_k)$ and multiply it by -1.

$$d(V_{t_i}, x, \boldsymbol{q}_k) = -d(V_{t_i}, x, -\boldsymbol{q}_k) \tag{3.3}$$

The use of this calculation result satisfies the single peak of $d(V_{t_i}, x, \boldsymbol{q}_k)$: As a voxel x included in V_{t_i} comes close to the surface of V_{t_i}, $d(V_{t_i}, x, \boldsymbol{q}_k)$ varies from positive to zero. On the contrary, as a voxel x which is not included in V_{t_i} comes close to the surface of V_{t_i}, $d(V_{t_i}, x, \boldsymbol{q}_k)$ varies from negative to zero.

These two facts show that $d(V_{t_i}, x, \boldsymbol{q}_k)$ is monotone increasing.

There is a case that there is no line-segment for $-\boldsymbol{q}_k$. In this case, we set $d(V_{t_i}, x, \boldsymbol{q}_k)$ to be ∞.

On the other hand, there is a case that the line-segment, which starts at the center of x and ends at surface of V_{t_i}, exists. One example of it is shown on voxel B's $\boldsymbol{q}_3$ in Figure3.2. $d(V_{t_i}, x, \boldsymbol{q}_k)$ is calculated by the another way when the line-segment exists. Suppose a case that $d(V_{t_i}, x, \boldsymbol{q}_k)$ is defined as Equation3.3. When a voxel x which is not included in V_{t_i} goes across the surface of V_{t_i}, $d(V_{t_i}, x, \boldsymbol{q}_k)$ has non-continuing change. The explanation of such change is given by end points of line-segments. A end point of line-segment acquired from a voxel which is not included in visual hull, which is shown in $\bar{b}_3$ on Figure3.2, is far from that included in visual hull, which is shown in b_3 on Figure3.2. To avoid the non-continuing change, we define $d(V_{t_i}, x, \boldsymbol{q}_k)$ as the length of the second shortest line-segment, not the shortest one (b_3 in Figure3.2).

All the $d(V_{t_i}, x, \boldsymbol{q}_k)(k = 1, \ldots, 6)$ for voxel A in Figure3.2 have positive value. $d(V_{t_i}, x, \boldsymbol{q}_2)$ and $d(V_{t_i}, x, \boldsymbol{q}_5)$ for voxel B have ∞ value, and $d(V_{t_i}, x, \boldsymbol{q}_1)$ and $d(V_{t_i}, x, \boldsymbol{q}_6)$ have negative value.

3.3.2 Dissimilarity

Let us focus on a voxel x which belongs to *part* j at t_i. At time t_0, the voxel occupied a cubic region. Transforming x with inverse-transformation of $D_j^{t_i}$ gives the region. We denote it by $\hat{x}_{D_j^{t_i}}$. The definition of dissimilarity

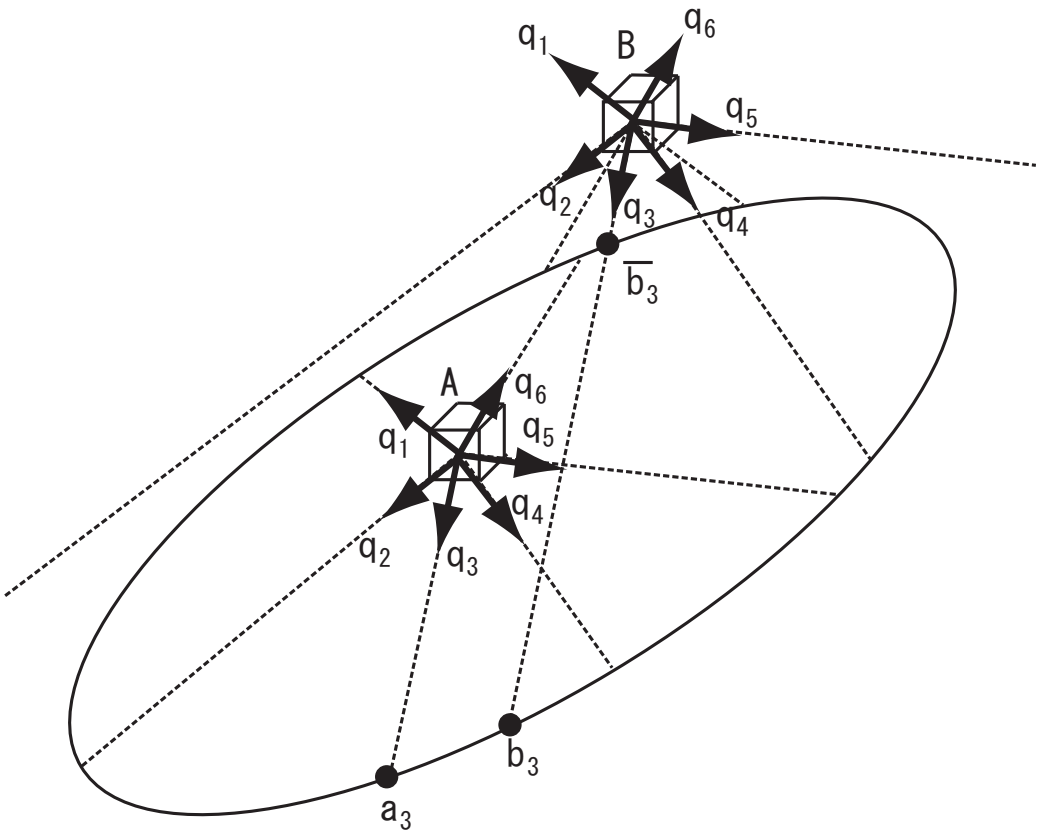


Figure 3.2: $d(V_{t_i}, x, \boldsymbol{q}_k)$

between x and $\hat{x}_{D_j^{t_i}}$ is given by

$$\mathfrak{D}(x, \hat{x}_{D_j^{t_i}}; V_{t_i}, V_{t_0}, D_j^{t_i}) = \mathfrak{S}^{\rho}_{\forall \boldsymbol{q}_k} \left\{ \left| d(V_{t_i}, x, \boldsymbol{q}_k) - d(V_{t_0}, \hat{x}_{D_j^{t_i}}, D_r^{-1}\boldsymbol{q}_k) \right| \right\} \tag{3.4}$$

where, D_r is a rotation matrix which $D_j^{t_i}$ includes, and $\mathfrak{S}^{\rho}_{\forall \boldsymbol{q}_k}$ is average over $s - \rho$ of s elements which are calculated in the following function.

$$\left| d(V_{t_i}, x, \boldsymbol{q}_k) - d(V_{t_0}, \hat{x}_{D_j^{t_i}}, D_r^{-1}\boldsymbol{q}_k) \right| \tag{3.5}$$

where, the $s - \rho$ elements are selected in ascending orger.

When $s-\rho$ of s $d(V_{t_i}, x, \boldsymbol{q}_k)$ are not effected by the unnecessary voxels, the dissimilarity has small difference during the motion and does not have large difference. We make an assumption that the difference is under a normal distribution.

In other words, even if ρ of s $d(V_{t_i}, x, \boldsymbol{q}_k)$ are effected by the unnecessary voxels, the dissimilarity keeps the stablability. The stablability means that the dissimilarity does not change during the motion. Of course, when more than ρ of s $d(V_{t_i}, x, \boldsymbol{q}_k)$ are effected by the unnecessary voxels, the dissimilarity will lose the stablability.

We conducted a simulation in order to determine ρ[27]. Three objects, which are shown in Figure3.3(a)(b)(c), are used for the simulation. Percentage of unnecessary voxels which are included in visual hull is measured.

The result of simulation is shown in Figure3.3(d). In this simulation, cameras are arranged on the vertices of the regular n-hedron. The simulation results show that the use of 20 cameras reduces the percentage of unnecessary voxels to 25%.

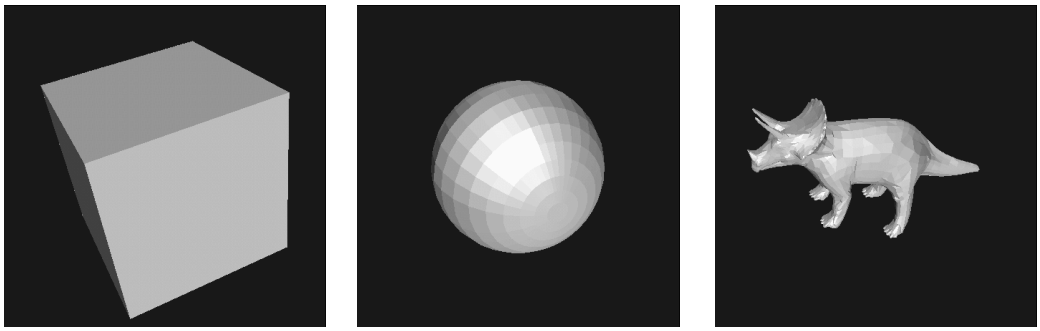
The result implies that 25% of s is reasonable number for ρ when we use 20 cameras.

3.4 Shape of Body Parts and Pose of Articulate Objects by EM Argorithm

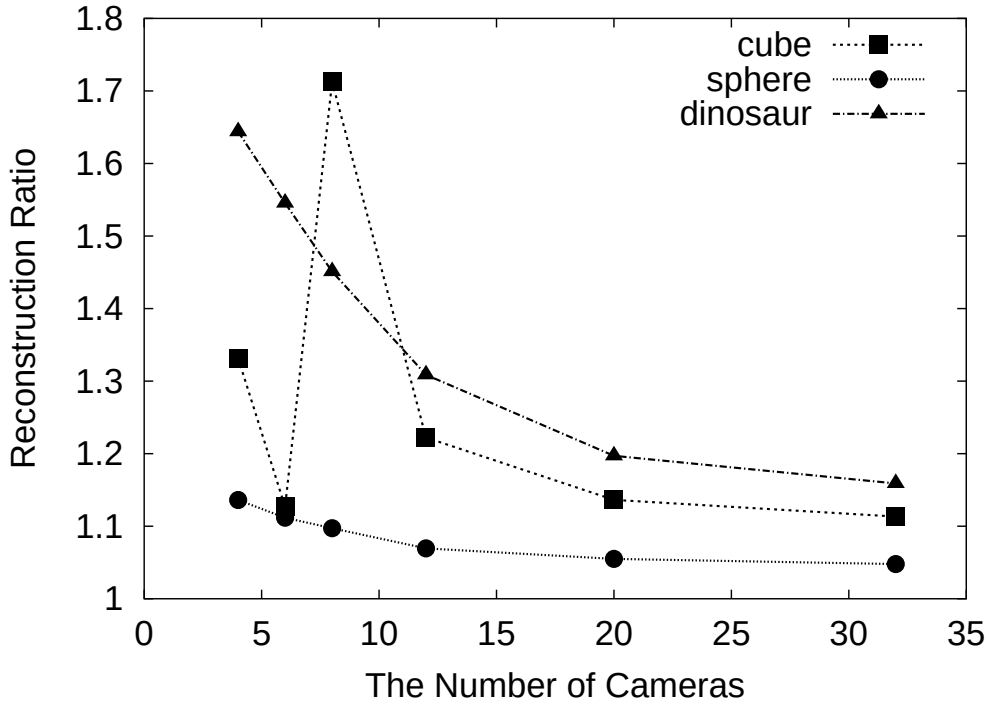
A set of label, $\boldsymbol{z}$, and rigid motion of each part, $D_j^{t_i}$ $(t_i = t_1, \ldots, t_N; j = 1, \ldots, M)$, are estimated by using V_{t_i} $(t_i = t_1, \ldots, t_N)$ and voxel feature and voxel dissimilarity. We estimate them with EM algorithm[41, 2, 42].

Joints of articulated objects have non-rigid motion. The non-rigid motion decreases an accuracy of estimation of $D_j^{t_i}$ $(t_i = t_1, \ldots, t_N; j = 1, \ldots, M)$. Probabilistic approaches solve the decrease. Our approach does not directly estimate the label of voxel x. Instead of this, it estimates $P(z(x) = j)$, which is a probability that a voxel x included in V_{t_0} belongs to *part* j. The use of voxels weighted by the probability gives more accurate rigid motion.

$P(z(x) = j)$ is estimated by two assumptions. The first assumption is concerning the dissimilarity described in 3.3. Change of the dissimilarity during the motion remains up to the size of voxel. The second assumption is that a voxel tends to belong to the same part that neighbor voxels belong to. We model it as Markov Random Fields (MRF)[9]. $P(z(x) = j|\boldsymbol{L}(x))$, which



(a) (b) (c)



(d)

Figure 3.3: the number of cameras vs volume of visual hull

is a conditional probability of $z(x) = j$ assuming that labels of 26-neighbor voxels are known , approximates $P(z(x)=j)$.

EM algorithm is a kind of the maximum likelihood estimation algorithm.It estimates parameters of complete-data from given incomplete-data. Our method uses the visual hull in time sequence V_{t_i} $(t_i = t_1, \ldots, t_N)$ and the labeling $\boldsymbol{z}$ as the complete-data. We can observe only the V_{t_i} $(t_i = t_1, \ldots, t_N)$, which are used as the incomplete-data.

EM algorithm consists of iterations of two steps; Expectation step(E-step) and Maximization step(M-step). It estimates parameters of probabilistic model of complete-data $\boldsymbol{\Xi}$ from given initial parameter $\boldsymbol{\Xi}^{(0)}$. In our method, the parameter $\boldsymbol{\Xi}$ consists of $D_j^{t_i}$ $(t_i = t_1, \ldots, t_N; j = 1, \ldots, M)$, σ_i and $\alpha_1, \ldots, \alpha_M, \beta$, which are described in the following section. Iterations of E-step and M-step give $\boldsymbol{\Xi}$, and also give $P(z(x) = j)$.

3.4.1 E-step

In the E-step, the expectation of a complete-data log-likelihood function $Q = E\left[l_c \middle| V_{t_1}, \ldots, V_{t_N}, \boldsymbol{\Xi}^{(p)}\right]$ is calculated, where l_c is a complete-data log-likelihood function. V_{t_i} $(t_i = t_1, \ldots, t_N)$ and current estimate of the parameters $\boldsymbol{\Xi}^{(p)}$ are given in the E-step. $P(z(x) = j)$ is updated in the E-step. Roughly speaking, in the E-step, the probability of voxel's label is estimated by using current estimate of $D_j^{t_i}$ $(t_i = t_1, \ldots, t_N; j = 1, \ldots, M)$.

First of all, we define l_c, which is included in Q. A probabilistic model of complete-data, which includes sequential visual hull and a set of label, gives l_c. l_c is defined as a mixture model of following two log function; one is a conditional log-likelihood of feature value $F_{V_{t_i}}$ assuming that $\boldsymbol{z}$ is given, the other is a log-likelihood of $\boldsymbol{z}$. The following equation gives us the definition of l_c.

$$l_c = \sum_{t_1}^{t_N} \left(\sum_{x \in V_{t_i}} \log P(F_{V_{t_i}}(x) | \boldsymbol{z}; \boldsymbol{\Xi}) + \log P(\boldsymbol{z}; \boldsymbol{\Xi}) \right) \tag{3.6}$$

The explanation of this definition is described as follows. l_c is the logarithmic of a probabilistic distribution of complete-data. The probabilistic distribution of complete-data consists of that of $\boldsymbol{z}$ and that of the visual hull. The probabilistic distribution of the visual hull consists of that of voxel feature included in V_{t_i}.

$P(F_{V_{t_i}}(x)|\boldsymbol{z};\boldsymbol{\Xi})$, which is in the right side of Equation3.6, is a probabilistic distribution of voxel feature $F_{V_{t_i}}(x)$ assuming that $\boldsymbol{z}$ is given. When voxel x belongs to *part* j, the dissimilarity between $F_{V_{t_i}}(x)$ and $F_{V_{t_0}}(\hat{x}_{D_j^{t_i}})$, which is denoted as $\mathfrak{D}(x, \hat{x}_{D_j^{t_i}}; V_{t_i}, V_{t_0}, D_j^{t_i})$, is under a normal distribution, and the dissimilarity remains up to the size of voxel. The definition of $\log P(F_{V_{t_i}}(x)|\boldsymbol{z};\boldsymbol{\Xi})$ is given in the following.

$$\begin{aligned}&\log P(F_{V_{t_i}}(x)|\boldsymbol{z};\boldsymbol{\Xi})\\ &= \log\left(\frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{\mathfrak{D}(x,\hat{x}_{D_j^{t_i}};V_{t_i},V_{t_0},D_j^{t_i})^2}{2\sigma_j^2}}\right)\end{aligned} \tag{3.7}$$

The following equation gives the conditional expectation of $\log P(F_{V_{t_i}}(x)|\boldsymbol{z};\boldsymbol{\Xi})$

$$\begin{aligned}&E\left[\log P(F_{V_{t_i}}(x)|\boldsymbol{z};\boldsymbol{\Xi})|V_{t_i},\boldsymbol{\Xi}^{(p)}\right] =\\ &\sum_j^M \lambda_j^{t_i(p)}(\hat{x}_{D_j^{t_i}})\left[\log P(F_{V_{t_i}}(x)|\boldsymbol{z};\boldsymbol{\Xi})\right]\end{aligned} \tag{3.8}$$

where, $\lambda_j^{t_i(p)}(\hat{x}_{D_j^{t_i}})$ is $P\left(\hat{x}_{D_j^{t_i}}|V_{t_i},\boldsymbol{\Xi}^{(p)}\right)$, which is a conditional expectation of $(\hat{x}_{D_j^{t_i}}|V_{t_i})$ assuming that V_{t_i} and $\boldsymbol{\Xi}^{(p)}$ are given.

$P(\boldsymbol{z};\boldsymbol{\Xi})$, which is in the right side of Equation3.6, is a priori probability of $\boldsymbol{z}$. We model the priori probability as the MRF. MRF models following behavior of the labels; a label of a voxel correlates with that of neighbor voxels, and it does not correlate with that of non-neighbor voxels.

MRF gives the priori probability of $\boldsymbol{z}$ in the following. Let $\delta(z(x)=j)$ be a binary function, and let $\tau_j(x) = \alpha_j + \beta\sum_{y\in\eta(x)}\lambda_j^{t_i(p-1)}(y)$, and $\eta(x)$ be 26-neighbor voxels of x. $\delta(z(x)=j)$ is equal to 1 only if the label of x is j, and is equal to 0 if the label of x is not j. $\alpha_1, \alpha_2, \ldots, \alpha_M, \beta$ are parameters of MRF. The priori probability of $\boldsymbol{z}$ is expressed as following equation.

$$P(\boldsymbol{z};\boldsymbol{\Xi}) = \prod_x \frac{\exp\left\{\sum_j^M \delta\left(\boldsymbol{z}(x)=j\right)\tau_j(x)\right\}}{\sum_k^M \exp\tau_k(x)} \tag{3.9}$$

The priori probability gives a conditional expectation of $\log P(\boldsymbol{z};\boldsymbol{\Xi})$

$$E\left[\log P(\boldsymbol{z};\boldsymbol{\Xi})|V_{t_i},\boldsymbol{\Xi}^{(p)}\right] = \sum_{x\in V_{t_0}}\sum_{j}^{M}\lambda_j^{t_i(p)}(x)\log\frac{\exp\tau_j(x)}{\sum_j^M\exp\tau_k(x)} \tag{3.10}$$

.

Approximation of $P(z(x)=j|\boldsymbol{\Xi}^{(p)})$ gives $\lambda_j^{t_i(p)}(x)$. $P(z(x)=j|\boldsymbol{L}(x),\boldsymbol{\Xi}^{(p)})$, which is a conditional probability of $z(x)=j$ assuming that labels of 26-neighbor voxel are given, approximates $P(z(x)=j|\boldsymbol{\Xi}^{(p)})$. $\lambda_j^{t_i(p)}(x)$ is acquired by using Bayes rule and $P(z(x)=j|\boldsymbol{L}(x),\boldsymbol{\Xi}^{(p)})$

$$\begin{aligned}\lambda_j^{t_i(p)}(x) &= P\left(z(x)=j|V_{t_i},\boldsymbol{\Xi}^{(p)}\right)\\ &= \frac{P(V_{t_i}|z(x)=j,\boldsymbol{\Xi}^{(p)})P(z(x)=j|\boldsymbol{\Xi}^{(p)})}{\sum_k P(V_{t_i}|z(x)=k,\boldsymbol{\Xi}^{(p)})P(z(x)=k|\boldsymbol{\Xi}^{(p)})}\\ &\approx \frac{P(V_{t_i}|z(x)=j,\boldsymbol{\Xi}^{(p)})\pi_j^{(p)}(x)}{\sum_k P(V_{t_i}|z(x)=k,\boldsymbol{\Xi}^{(p)})\pi_k^{(p)}(x)}\end{aligned} \tag{3.11}$$

where we denote $P\left(z(x)=j|\boldsymbol{L}(x),\boldsymbol{\Xi}^{(p)}\right)$ as $\pi_j^{(p)}(x)$.

Approximation $P(\boldsymbol{z};\boldsymbol{\Xi})\approx\prod_x P(z(x)|\boldsymbol{L}(x),\boldsymbol{\Xi})$ [41] and Equation3.9 give

$$P(z(x)|\boldsymbol{L}(x),\boldsymbol{\Xi}) = \frac{\exp\left\{\sum_j^M\delta\left(z(x)=j\right)\tau_j(x)\right\}}{\sum_k^M\exp\tau_k(x)} \tag{3.12}$$

.

Q is acquired from Equation3.6, 3.8, 3.10, 3.11, and 3.12.

3.4.2 M-step

M-step estimates $\boldsymbol{\Xi}$ which maximizes Q. The estimated $\boldsymbol{\Xi}$ replaces the current estimate $\boldsymbol{\Xi}^{(p)}$ by itself. Roughly speaking, in the M-step, rigid motion of each part $D_j^{t_i}$ $(t_i = t_1,\ldots,t_N; j=1,\ldots,M)$ is estimated by using the probability of voxel's label.

Maximization of Equation3.8, which is a component of Q, gives the estimate of $D_j^{t_i}$. Voxels in non-rigid region have small $\lambda_j^{t_i(p)}(x)$. The small $\lambda_j^{t_i(p)}(x)$ has small contribution for $D_j^{t_i}$ estimation. The small contribution reduces the effect of non-rigid motion.

3.4.3 Initial Estimate of Probability

EM algorithm requires the initial estimate of Ξ and probability of the label of voxels $\lambda_j^{t_i(p)}(x)$. It also requires the number of parts M.

We describe the method for estimating these parameters. Our method divides V_{t_0} into small regions, and estimates the rigid motion of each region. The estimation gives rough shape and motion of each part.

V_{t_0} division

V_{t_0} is divided into small regions, which we call voxel-block. Each voxel-block satisfies three conditions. One condition is that voxels in a voxel-block should be distributed within the fixed-sized cubic area. Let r to be the size of cubic area. Second condition is that a voxel-block should be single connected region. Third condition is that each part should contain at least one voxel-block. These conditions determine the size r.

Rigid motion estimation

We assume that each voxel-block has rigid motion and estimate the rigid motion of each voxel-block. The estimation is done in the following order; motion between t_0 and t_1, t_0 and t_2,..., t_0 and t_n.

The rigid motion between t_0 and t_i is estimated by using template matching method and initial estimate of rigid motion. The rigid motion between t_0 and t_{i-1} is used as the initial estimate. The template matching method uses voxel feature described in section3.3, and uses the voxel-block as a template, and matches the template to V_{t_i}.

The template matching method minimizes a matching function, which is the sum of $\mathfrak{D}$ for each voxel included in a voxel-block,

$$E = \sum_{x \in voxelblock} \mathfrak{D}(x, \hat{x}_{D_j^{t_i}}; V_{t_i}, V_{t_0}, D) \tag{3.13}$$

and rigid motion D which minimizes a matching function is acquired.

Clustering

Some voxel-blocks may include voxels which are included in different parts. The motions of such voxel-blocks are not rigid motion. When we try to

estimate the rigid motion of such voxel-blocks, minimized E shown in Equation3.13 has large value because of the non-rigidity of voxel-blocks.

In order to detect such non-rigid voxel-blocks, we extract voxel-blocks whose minimized E are under the given threshold.

Some voxel-blocks may have same rigid motion. We use a clustering method in order to group the voxel-blocks which have same rigid motion into a single cluster.

The number of the acquired clusters, let them to be $cluster_j (j = 1, \ldots, M)$, gives the number of body part M.

$\lambda_j^{t_i(0)}(x)$ is initialized as follows: When x is included in $cluster_k (k \neq j)$, $\lambda_j^{t_i(0)}(x)$ has 0. When x is included in $cluster_j (k \neq j)$, $\lambda_j^{t_i(0)}(x)$ has 1. When x is not included in any cluster, $\lambda_j^{t_i(0)}(x)$ has $\frac{1}{M}$.

We set the initial estimate of $D_j^{t_i}$ to be D, and σ_j to be the square means of $\mathfrak{D}(x, \hat{x}_D; V_{t_i}, V_{t_0}, D)$.

3.4.4 Segmentation

As a result of the EM algorithm, different parts may have same rigid motion. The one explanation of this is that estimated M may be larger than the number of parts. The clustering algorithm described in 3.4.3 fixes the excessive M. Parts whose rigid motions are clustered into same rigid motion are merged into single part.

Voxels whose $\pi_j^{(p)}(x)$ are larger than a given threshold th gives the shape of part j.

3.5 Experiment

The experimental results are shown and are discussed in this section. We conducted two experiments; experiments with synthetic data and that with real data.

3.5.1 Experiment with Synthesis Data

A cow model shown in Figure3.4 is used for the experiment. The cow model has five parts; a body and four legs. Ten frames of walking motion data observed by 20 cameras are used as input image sequences. The walking motion has four different rotations: each leg rotates backwards and forwards.

Figure3.4 shows sequential visual hull reconstructed from the input. Each visual hull has approximately $100 \times 170 \times 270$ voxels.

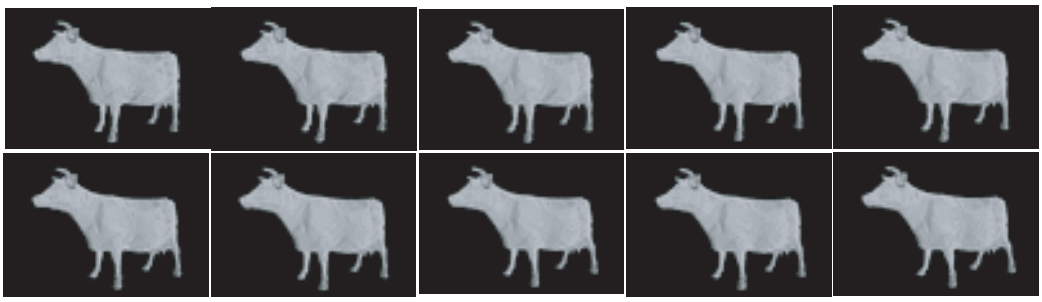


Figure 3.4: View Volume

Figure3.5(c) shows the acquired part with our method. Figure3.5(d) illustrates the same result as Figure3.5(d) and zooms in a joint between the body and the right-front leg.

We set $s = 20, \rho = 5, r = 16$ and extracted voxels whose $\pi_j^{(p)}(x)$ is larger than $th = 0.9$. Our method determined M with 5 after the reduction described in 3.4.4. Figure3.5(c) shows that the shapes of five parts of the cow model are acquired by using our method.

The accuracy of estimated motion is evaluated. The difference between the estimated $D_j^{t_i}$ and the part's motion, which was given when we created the model's motion, gives the accuracy. In order to evaluate the effectiveness of using weighted voxels, two methods are tested for estimating the accuracy; one is our method with weighted voxels, the other is same as our method except for using non-weighted voxels. The latter method employs the following equation instead of Equation3.8

$$E\left[\log P(F_{V_{t_i}}(x)|\boldsymbol{z};\boldsymbol{\Xi})|V_{t_i},\boldsymbol{\Xi}^{(p)}\right] = \sum_j^M \bar{\lambda}_j^{t_i(p)}(\hat{x}_{D_j^{t_i}})\left[\log P(F_{V_{t_i}}(x)|\boldsymbol{z};\boldsymbol{\Xi})\right] \tag{3.14}$$

,where $\bar{\lambda}_j^{t_i(p)}(\hat{x}_{D_j^{t_i}})$ is a binary function.

$$\bar{\lambda}_j^{t_i(p)}(\hat{x}_{D_j^{t_i}}) = \begin{cases} 1 & \text{if } \bar{\lambda}_j^{t_i(p)}(\hat{x}_{D_j^{t_i}}) \geq \bar{\lambda}_k^{t_i(p)}(\hat{x}_{D_k^{t_i}}) \text{ for } \forall k \neq j \\ 0 & \text{else} \end{cases}$$

Table 3.1: Average error by pose estimation.

	rotation error(deg)		translation error(vox)	
weighting	yes	no	yes	no
body	0.40	0.35	1.25	1.57
left front leg	1.59	3.41	1.43	2.33
right front leg	1.35	5.52	2.02	3.20
left back leg	1.24	2.92	1.54	2.06
right back leg	1.40	2.31	0.89	2.00

The accuracy of estimated motion is shown in Table3.1. Two types of error are calculated: One is the average error of rotation, and the other is that of translation. Both errors are averaged over the observed time. The rotation errors are given by an rotate-angle of R_{err},

$$R_{err} = R\hat{R}^T \tag{3.15}$$

where $\hat{R}$ is an estimated rotation, and R is a correct rotation.

Table3.1 indicates that our method could estimate the motion accurately. Our method estimated the rotation with less than 2 degrees error and the translation with less than 2 voxels error. On the contrary, the method without the weighting increased the errors; 5 degrees for rotation, 3 voxels for translation.

Figure3.5(e)(f) also illustrates the accuracy of motion estimation. The figures illustrate two volume data; one is V_{t_9}, visual hull of last frame, the other is transformed V_{t_0}. The transformed V_{t_0} contains parts whose voxels are transformed by $D_j^{t_9}$. Figure3.5(f) illustrates the same result as Figure3.5(e) and zooms in a joint between the body and the right-front leg. V_{t_9} is illustrated as white translucent area. Transformed parts are colorized as red, blue, yellow, green, and cyan region. The cyan region covers the right-front leg of V_{t_9}, however, it does not cover the joint between the body and the right-front leg. Except for such region, the rotation with less than 2 degrees error and the translation with less than 2 voxels error are enough to reconstruct the motion of articulate object.

Figure3.5(g) also illustrates the two volume data. The difference with Figure3.5(f) is that transformed V_{t_0} on Figure3.5(g) is acquired from non-weighted voxels.

Comparison between Figure3.5(f) and (g) indicates the effectiveness of using weighted voxels. Figure3.5(g) illustrates V_{t_9} as white translucent area and transformed right-front leg as cyan area. A red circle on Figure3.5 indicates that transformed right-front leg overwraps V_{t_9}. A yellow circle on Figure3.5 indicates that V_{t_9} overwraps the transformed right-front leg. These circles show that the use of non-weighted voxels decreases the accuracy of motion estimation.

3.5.2 Experiment with Real Data

A hand shown in Figure3.7 is used for the experiment. Ten frames of finger bending motion observed by 20 cameras are used as input image sequences. The bending motion has four different rotations: four of five fingers bend inward. Figure3.7 shows sequential visual hull reconstructed from the input. We set the size of voxel $1 \times 1 \times 1$(mm).

Figure3.7(b) shows the acquired part with our method. We set $s = 20, \rho = 5, r = 16$ and extracted voxels whose $\pi_j^{(p)}(x)$ is larger than $th = 0.9$. Our method determined M with 5 after the reduction described in 3.4.4. Figure3.7(b) shows that the shape of four fingers are acquired by using our method.

Figure3.7(d) illustrates the accuracy of motion estimation.The figures illustrate two volume data; one is V_{t_9}, visual hull of last frame, the other is transformed V_{t_0}. The transformed V_{t_0} contains parts whose voxels are transformed by $D_j^{t_9}$. Figure3.7(e) is a view of the two volume data from a different viewpoint. V_{t_9} is illustrated as white translucent area. These figures show that our method successfully reconstructs the hand motion.

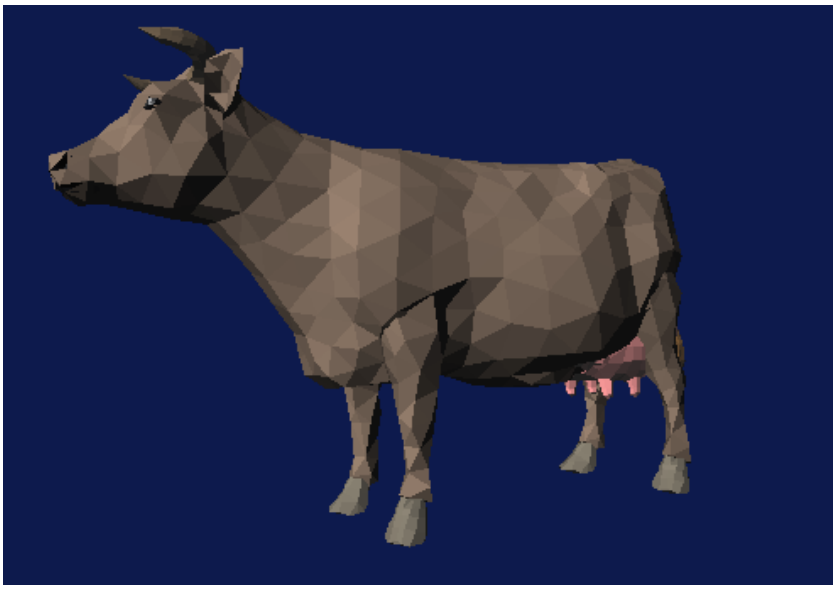
3.6 Conclusion

In this chapter, we proposed a method to acquire both the shape of body parts and the pose of an articulated object without using color information.

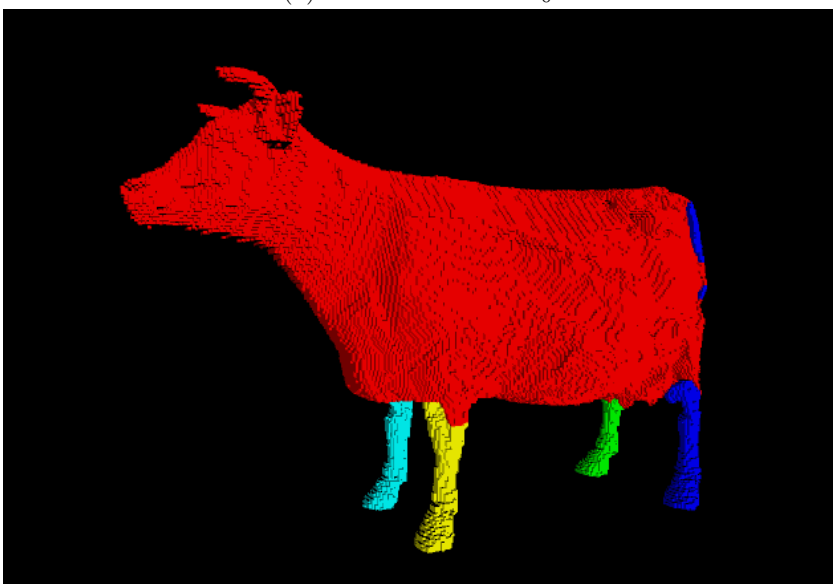
Our method employs a multi-dimensional distance in order to avoid the bad effects from the unnecessary voxels, which are included in the visual hull. Our method avoids the bad effects from the non-rigid motion, employing the probabilistic approach.

Acquisition of non-rigid motion is one of the future works. Our method can produce an appearance with an arbitrary pose but produces the appear-

ance around non-rigid region with low accuracy. The use of a meta-ball representation or superquadric function is a possible solution for the non-rigid motion acquisition.

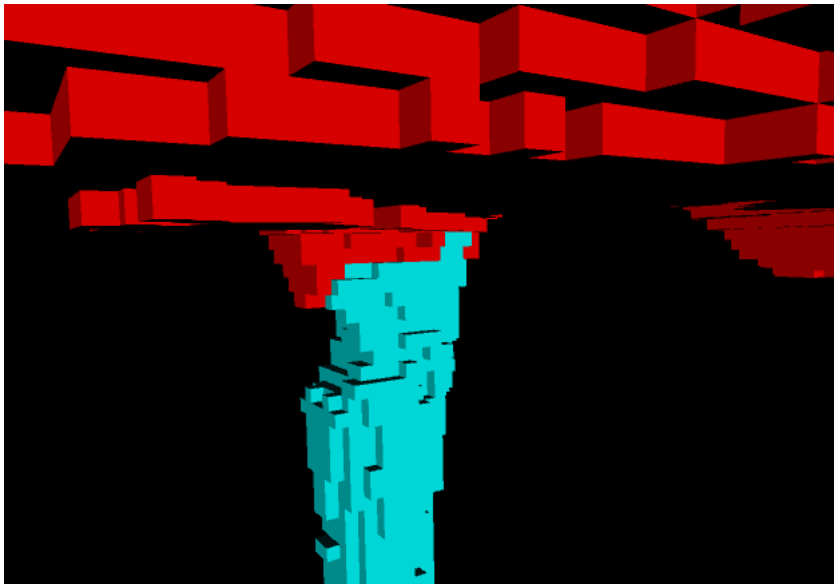


(a) 3D cow model at t_0



(b) Segmented result

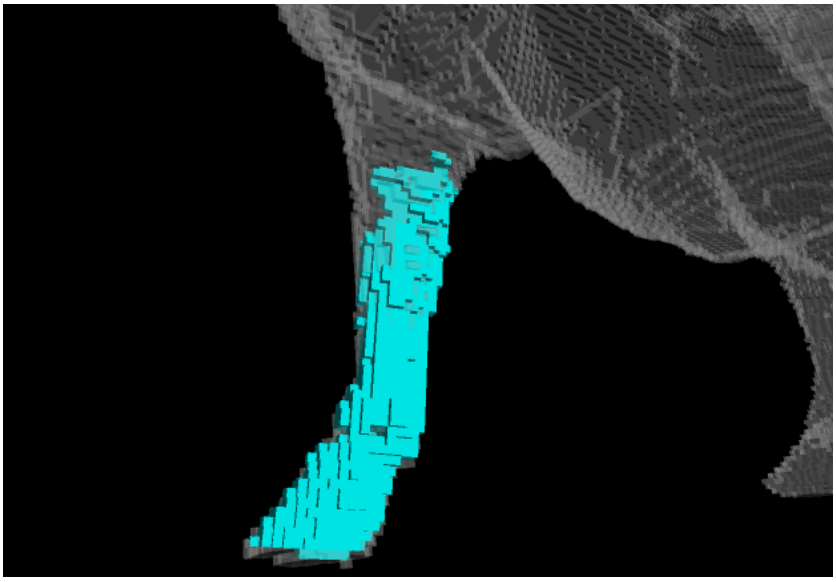
Figure 3.5: Experimental results with synthesis data



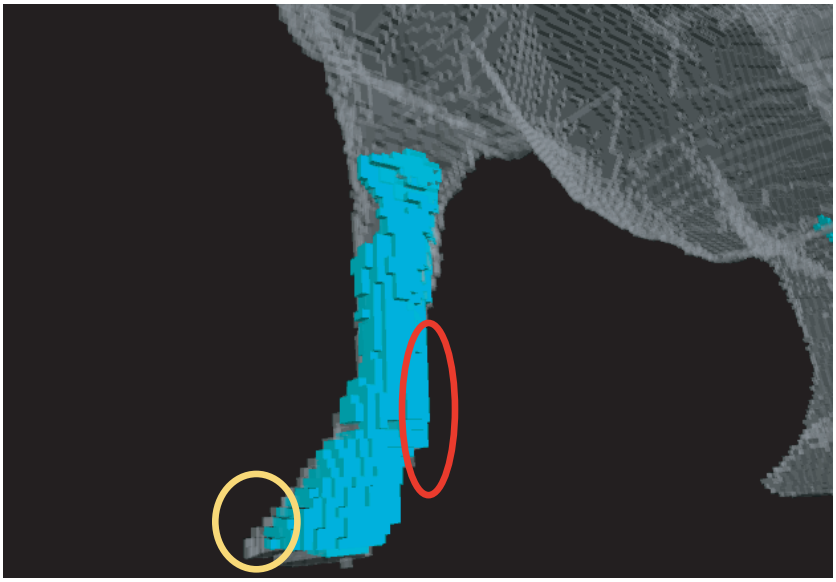
(c) Segmented result(zoom)

(d) Motion after the segmentation

Figure 3.5: Experimental results with synthesis data



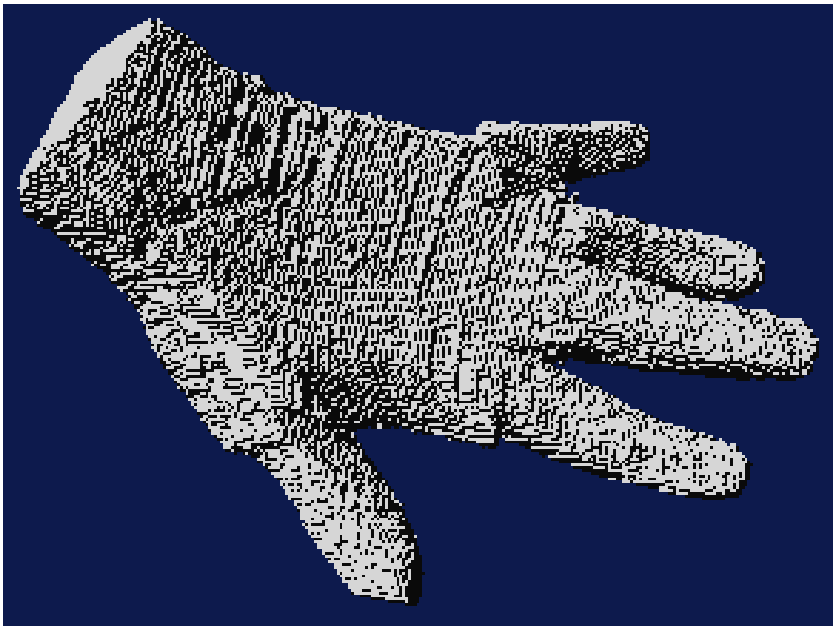
(e) Motion after the segmentation(zoom))



(f) Motion after the segmentation(zoom))

Figure 3.5: Experimental results with synthesis data

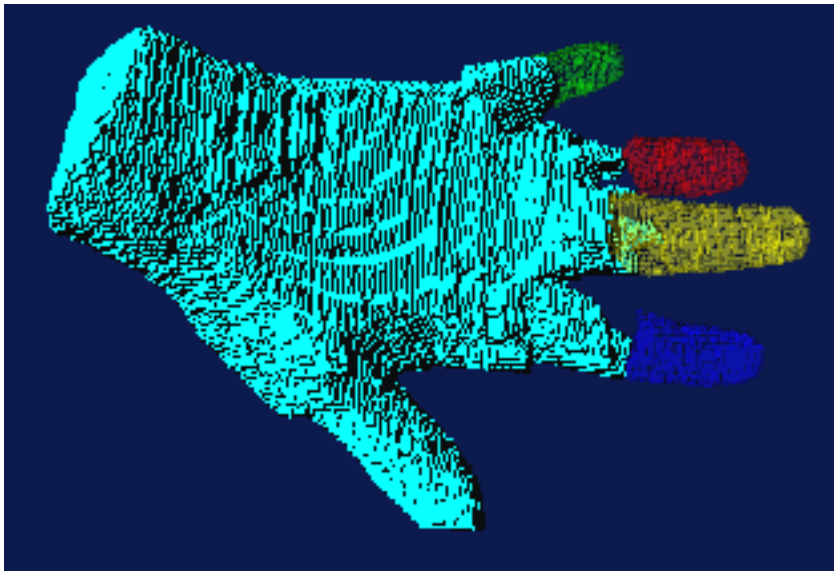
Figure 3.6: image sequence



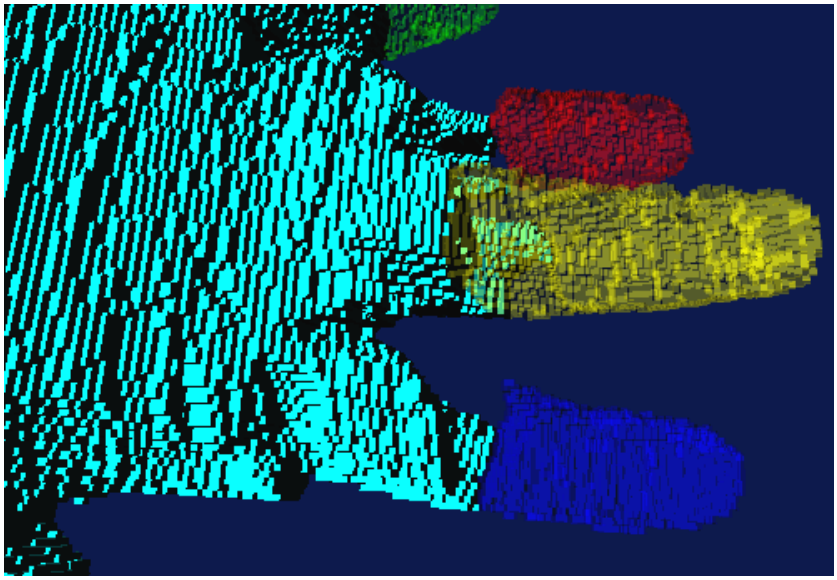
(a) View Volume Vt_0

(b) View Volume Vt_9

Figure 3.7: Results by hand shape data.

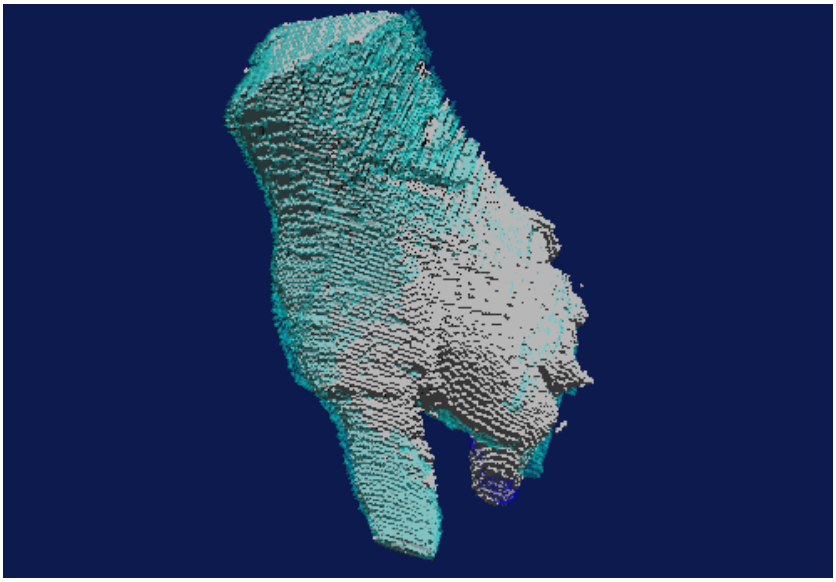


(c) Segmented result

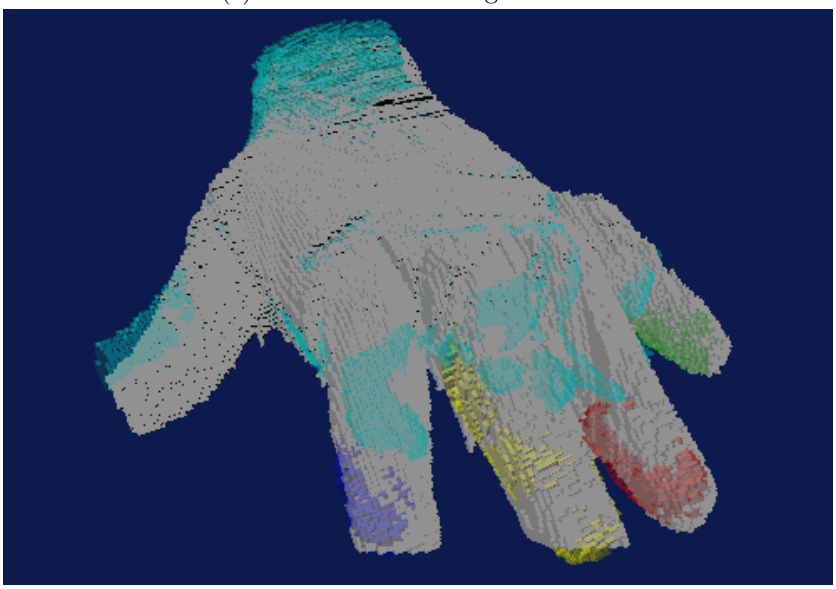


(d) Segmented result(zoom)

Figure 3.7: Results by hand shape data.



(e) Motion after the segmentation



(f) Motion after the segmentation (other view)

Figure 3.7: Results by hand shape data.

Chapter 4

Shape Acquisition with Controlable Lighting Environments

A method which reconstructs smooth or concave surface, which the volume intersection method can not reconstruct, is proposed. The method reconstructs depth maps from needle maps obtained by photometric stereo, controling a lighting environment. The needle maps contain the surface normals of smooth or concave surfaces. In case when the needle maps contain depth edges, incorrect depth maps are reconstructed, however. In order to fix the incorrect depth maps, we use silhouettes observed from various viewpoints. An incorrect depth map is not consistent with silhouettes which are taken from other viewpoints. Based on this fact, our method minimizes two types of energy functions to reconstruct the depth map: One energy function is based on a consistency between depth map and needle map, and the other is based on a consistency between depth map and silhouettes.

4.1 Introduction

The volume intersection method has an advantage for acquiring the shape of texture-less objects, but it can not acquire concave surface of the objects. It requires a lot of cameras in order to acquire smooth surface.

Methods using laser or pattern light[36] have been proposed to acquire the concave surface, but they require high-cost equipments. Image based

methods, including multi-baseline stereo[32] and space carving[19], can acquire the concave surfaces without high-cost equipments. They, however, use color information and can not acquire the shape of texture-less objects.

Photometric stereo[35, 3] estimates a surface normal as a needle map, and the concave and smooth surface, which the volume intersection method can not acquire, can be acquired from the surface normal. It uses a set of images acquired from single camera, and images are acquired under at least three different lighting-direction. It only requires low-cost equipments – a camera and several lights –, and it can acquire the shape of texture-less objects. Using the needle map, we acquire the concave surface.

Maki et al[25] proposed a method for reconstructing 3D surface of texture-less object. Their method requires that the object should be under a rigid motion, and it also requires that the motion should be estimated with Structure-from-motion; it is not easy to estimate the motion for texture-less object.

Cho[5] and Chen[4] had proposed approaches which uses photometric stereo. They used multiple cameras and acquired the needle maps from each camera. The distance map was reconstructed by using the consistency between the needle map and a surface normal calculated from the distance map. Distance maps reconstructed from the needle maps were merged into 3D space by their approaches. The merge was done by searching depth offsets in order to match the depth map to silhouettes.

Depth edges[34] make it difficult to acquire the distance map. A depth edge is an area on the distance map: On the area, a depth from a camera to the object's surface varies discontinuously. The discontinuity disables calculation of surface normal. It means that existence of depth edges disables a calculation of the consistency with the needle map and wrong distance map is acquired.

Silhouettes taken from different viewpoints reduce the bad effects of the depth edge. The wrong distance map will not match with a silhouette from other camera. Finding the mismatch detects the depth edges and corrects the wrong shape. In other words, consistency between the depth image and the silhouettes reduces the bad effects of depth edge.

We define two energy functions; an energy function based on the consistency with the needle map and that based on the consistency with the silhouettes. Minimizing these two energy functions provides a distance map which keeps the consistencies with the needle maps and the silhouettes, and it reduces the bad effects of the depth edge.

4.2 Photometric Stereo

Our method uses multiple cameras. Let the number of cameras to be C and let a silhouette on camera c ($c = 1, \ldots, C$) to be S_c.

The volume intersection method described in Chapter2 has an advantage over other shape acquiring methods. The advantage is that the method can acquire the shape of texture-less object. The volume intersection method requires object's silhouettes and does not require point correspondence, which other methods require. Extracting silhouettes of texture-less is easier task than obtaining point correspondence.

The volume intersection method has also a disadvantage. The disadvantage is the difficulty of acquiring smooth and concave surfaces. The visual hull, which is acquired by the volume intersection method, is convex hull circumscribing the object. Acquiring concave surface with the volume intersection method is impossible. The volume intersection method requires many cameras in order to acquire a smooth surface, even if the surface is not a concave surface.

We employ photometric stereo in order to acquire the smooth and concave surfaces.

Photometric stereo estimates surface normals of an object as a needle map. It uses a set of images acquire with single camera. Each image is acquired under a different lighting-direction. Images acquired under at least three different lighting-directions enable the needle map estimation.

The needle map contains surface normals of concave and smooth surfaces. We acquire the needle maps from each camera in order to acquire the shape.

Photometric stereo requires following assumptions.

- Each light is a directional light(or a point light located far from the object).
- Directions of all lights are known and at least three directions of them are linearly independent.
- No cast-shadow is observed.
- The object's surface is modeled as lambertian surface.

Basic idea and algorithm of photometric stereo is described as follows. A light reflected on the object's surface is observed by a camera. Let direction of the lights to be $\mathbf{l}_j^c (j = 1, \ldots, N_c)$. $\mathbf{l}_j^c$ is expressed as a 3D vector as shown

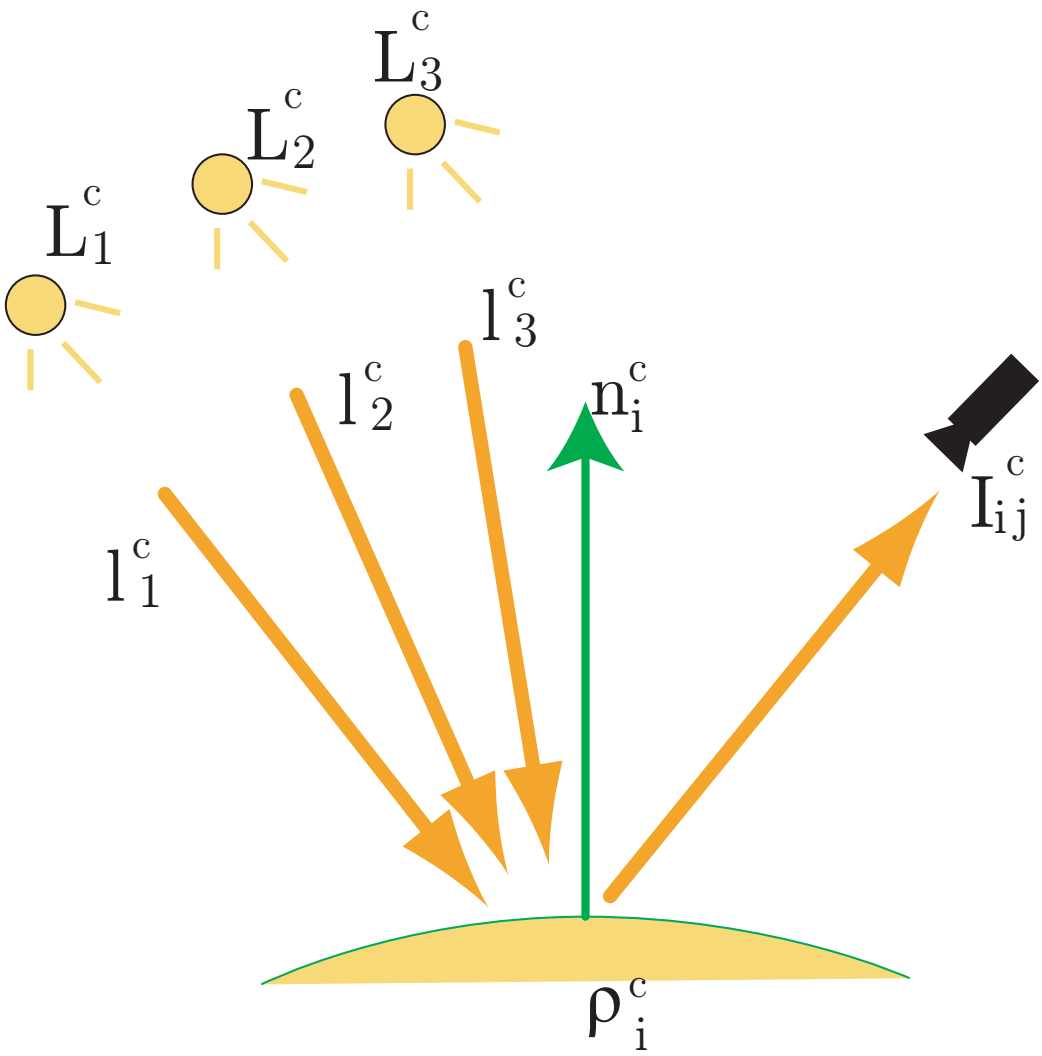


Figure 4.1: Photometric Stereo

in Figure4.1 The light whose direction is $\mathbf{l}_j^c$ diffuses on the object's surface, and the diffused light is observed by the camera. Pixels on the image of the camera represent the strength of the diffused light. Let the number of pixels on S_c to be M_c and let each pixel to be described by $m_i^c (i = 1, \ldots, M_c)$. A pixel m_i^c observes the diffused light from $\mathbf{l}_j^c$ and gets a pixel value I_{ij}^c. The following equation gives I_{ij}^c.

$$I_{ij}^c = L_j^c \rho_i^c \frac{\mathbf{l}_j^c \cdot \mathbf{n}_i^c}{\left|\mathbf{l}_j^c\right| \left|\mathbf{n}_i^c\right|} \tag{4.1}$$

where, L_j^c is a strength of the incident light, $\mathbf{n}_i^c$ is a normal vector of a surface observed by m_i^c, and ρ_i^c is a diffuse reflection factor of the surface.

Let us supporse $M_c \times N_c$ matrix $\mathbb{I}$ whose elements are I_{ij}^c. Equation4.1 derives that the matrix $\mathbb{I}$ consists two matrices; $M_c \times 3$ matrix $\mathbb{N}$ which includes the surface normals,and $3 \times N_c$ matrix $\mathbb{L}^{\mathrm{T}}$ which includes light directions. Decomposition $\mathbb{I}$ into $\mathbb{N}$ and $\mathbb{L}^{\mathrm{T}}$ gives $\mathbf{n}_i^c$, a normal vector of a surface observed by m_i^c. Singular Value Decomposition and known $\mathbf{l}_j^c$ give the decomposition.

$$\begin{aligned}
\mathbb{I} &= \begin{pmatrix} I_{11}^c & \cdots & I_{1N_c}^c \\ \vdots & \ddots & \vdots \\ I_{M_c1}^c & \cdots & I_{M_cN_c}^c \end{pmatrix} = \mathbb{N}\mathbb{L}^{\mathrm{T}} \\
&= \begin{pmatrix} \rho_1^c \frac{\mathbf{n}_1^{c\mathrm{T}}}{\left|\mathbf{n}_1^c\right|} \\ \vdots \\ \rho_{M_c}^c \frac{\mathbf{n}_{M_c}^{c\mathrm{T}}}{\left|\mathbf{n}_{M_c}^c\right|} \end{pmatrix} \begin{pmatrix} L_1^c \frac{\mathbf{l}_1^c}{\left|\mathbf{l}_1^c\right|} & \cdots & L_{N_c}^c \frac{\mathbf{l}_{M_c}^c}{\left|\mathbf{l}_{M_c}^c\right|} \end{pmatrix}
\end{aligned} \tag{4.2}$$

Photometric stereo has the same advantage as the volume intersection has. The advantage is that photometric stereo does not use color consistency and it can acquire the needle map of texture-less object.

4.2.1 Distance Map Reconstruction from Needle Map

The needle map acquired by photometric stereo does not directly express a shape of the object. It only expresses the surface normal. Acquiring the shape requires reconstruction of a distance map from the needle map.

Maximizing the following consistency gives the distance map. The consistency is that the needle map is consistent with the surface normal derived from the distance map. We call the consistency needle map consistency.

Depth edges make it difficult to calculate the consistency. A depth edge is an area on the distance map: On the area, a depth from camera to the surface varies discontinuously. The discontinuity disables calculation of surface normal; it means that existence of depth edges disables a calculation of the needle map consistency.

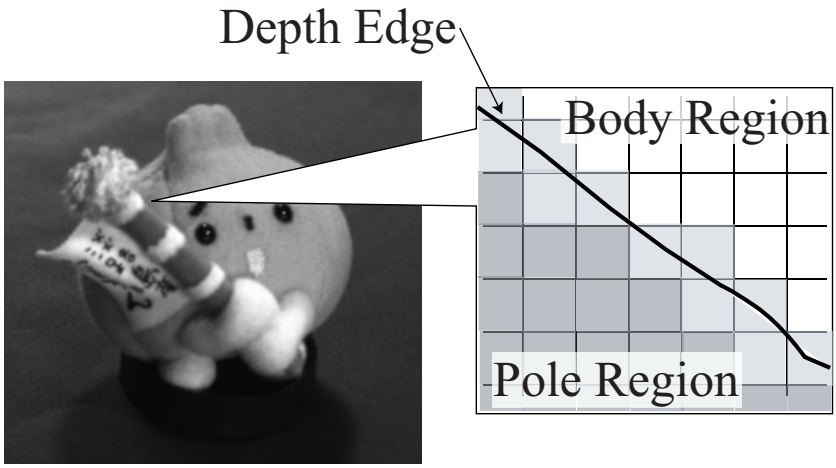
A depth edge is shown on Figure4.2(a). on an area between body and bar of a mascot, a depth from camera to the surface varies discontinuously. The surface normal can not be calculated on such area. Let suppose a pixel in the needle map where the body and the bar of the mascot are adjacent. Two disconnected surface are projected in the pixel, and it increases the error on estimated surface normal in the pixel. Such surface normal should not be used for calculating the needle map consistency.

The use of incorrect surface normals makes a wrong shape. A shape on which the body and the bar of a mascot are connected smoothly will be acquired (Figure4.2).

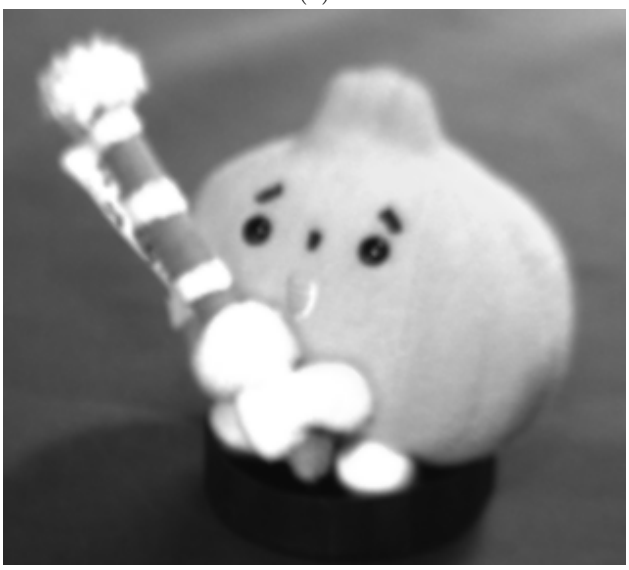
4.3 Shape Reconstruction by Using Silhouette and Needle Map Consistency

Silhouettes taken from different viewpoints reduce the bad effects of the depth edges. The following example provides the explanation. Let us suppose a silhouette taken from different camera (Figure4.2(b)). The body and bar of the mascot, which are adjacent on Figure4.2(a), are not adjacent on Figure4.2(b). Reconstructed shape from Figure4.2(a), on which the body and the bar of the mascot are connected smoothly, will not match with Figure4.2(b). Finding the mismatch detects the depth edges and corrects the wrong shape. In other words, consistency between the depth image and the silhouettes reduces the bad effects of depth edge. We call the consistency silhouette consistency.

We define two energy functions; an energy function based on needle map consistency, and that based on silhouette consistency. Minimizing these two energy functions provides a distance map which keeps the consistencies with the needle maps and that with the silhouettes , and it reduces the bad effects of depth edge.



(a)



(b)

Figure 4.2: Depth edge

Before discussing the definition of these energy functions, let us explain a pixel m_i^c and a distance map. A pixel m_i^c included in a silhouette S_c occupies the square region on the image. We describe it $([x_i^c, x_i^c + 1), [y_i^c, y_i^c + 1))$ and call a point (x_i^c, y_i^c) representative point of m_i^c. A depth of m_i^c is defined as a distance between focal point of the camera c and a point on the surface projected on the representative point of m_i^c. We express the depth as $Z(x_i^c, y_i^c)$. A distance map is a 2D matrix which contains the distances of each representative points.

4.3.1 Needle Map Consistency

We first discuss the needle map consistency, consistency between the distance map and the needle map.

Figure4.3 illustrates a depth of a pixel m_i^c, which is denoted by $Z(x_i^c, y_i^c)$, and a surface normal $\mathbf{n}_i^c = (p_i^c, q_i^c, 1)^{\mathrm{T}}$, which is observed by the pixel m_i^c.

Let us suppose that the surface observed by m_i^c is a plane containing surface normal $\mathbf{n}_i^c$. Depths of three points, $(x_i^c + 1, y_i^c)$,$(x_i^c, y_i^c + 1)$ and $(x_i^c + 1, y_i^c + 1)$, are given by $\mathbf{n}_i^c$ and $Z(x_i^c, y_i^c)$. When we express the depths $Z_{\substack{-1\\ \pm 0}}$, $Z_{\substack{\pm 0\\ -1}}$, $Z_{\substack{-1\\ -1}}$, they can be written

$$Z_{\substack{-1\\ \pm 0}}(x_i^c + 1, y_i^c) = Z(x_i^c, y_i^c)\left(1 + \frac{p_i^c}{f_c}\right) \tag{4.3}$$

$$Z_{\substack{\pm 0\\ -1}}(x_i^c, y_i^c + 1) = Z(x_i^c, y_i^c)\left(1 + \frac{q_i^c}{f_c}\right) \tag{4.4}$$

$$Z_{\substack{-1\\ -1}}(x_i^c + 1, y_i^c + 1) = Z(x_i^c, y_i^c)\left(1 + \frac{p_i^c}{f_c} + \frac{q_i^c}{f_c}\right) \tag{4.5}$$

where f_c is a focal length of the camera c.

These depths show that we have four ways of acquiring the depth of (x_i^c, y_i^c); $Z(x_i^c, y_i^c)$, a depth from $(x_i^c - 1, y_i^c)$ with Equation4.3, a depth from $(x_i^c, y_i^c - 1)$ with Equation4.4, and a depth from $(x_i^c - 1, y_i^c - 1)$ with Equation4.5. These ways give the needle map consistency. When the four depths are close together, they provide a high consistency. It gives the following energy expressing the needle map consistency. Lower energy shows the higher consistency.

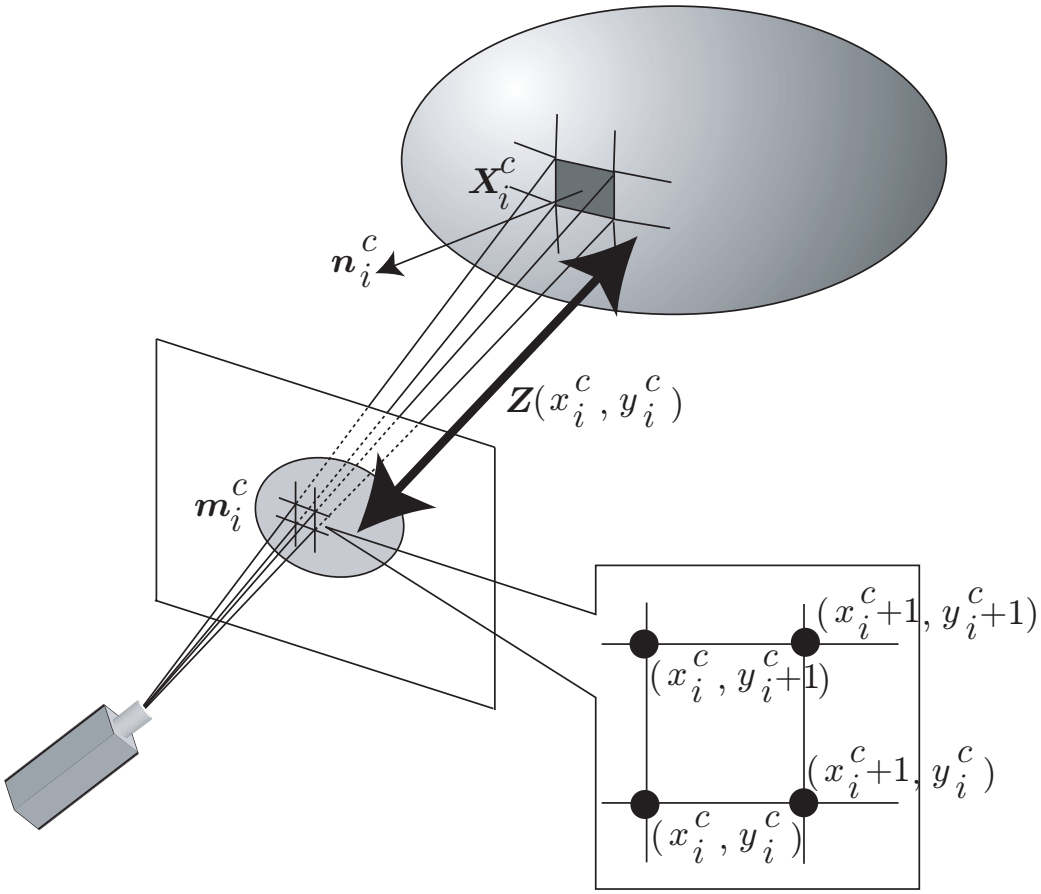


Figure 4.3: Needle map vs surface normal

$$E_N = \sum_{m_i^c \in S_c} \left(\left| Z(x_i^c, y_i^c) - Z_{\substack{-1 \\ \pm 0}}(x_i^c, y_i^c) \right|^2 + \left| Z(x_i^c, y_i^c) - Z_{\substack{\pm 0 \\ -1}}(x_i^c, y_i^c) \right|^2 + \left| Z(x_i^c, y_i^c) - Z_{\substack{-1 \\ -1}}(x_i^c, y_i^c) \right|^2 \right) \tag{4.6}$$

4.3.2 Silhouette Consistency

The silhouette consistency, consistency between the depth map and the silhouettes, is discussed.

The volume intersection method is one of the methods which use the silhouette consistency. The silhouettes of the object completely correspond with silhouettes of the object's visual hull. The correspondence gives the silhouette consistency.

Visual Hull Line

Some pixels included in a silhouette are adjacent to pixels which are not included in the silhouette. We call such pixels edge pixels. Pixels containing at least one of 8-neighbor pixels which is not included in the silhouette are extracted by our method.

Suppose a view-line which starts from the focal point of a camera and passes through a representative point of the edge pixel (Figure4.4).

Projecting the view-line to the other camera gives a 2D line on the camera's image. The 2D line intersects a silhouette on this camera's image. That is, some parts of the 2D line are included in the silhouette. In other words, some parts of the view-line are projected into the silhouette. A part of the view-line which is projected into all the silhouettes is called a visual hull line. Ignoring a sampling error, we consider that the visual hull lines are located on the surface of the visual hull.

The silhouettes of the object completely match with silhouettes of the object's visual hull. This fact and the visual hull line give the following constraints.

- The object never intersects any visual hull lines.

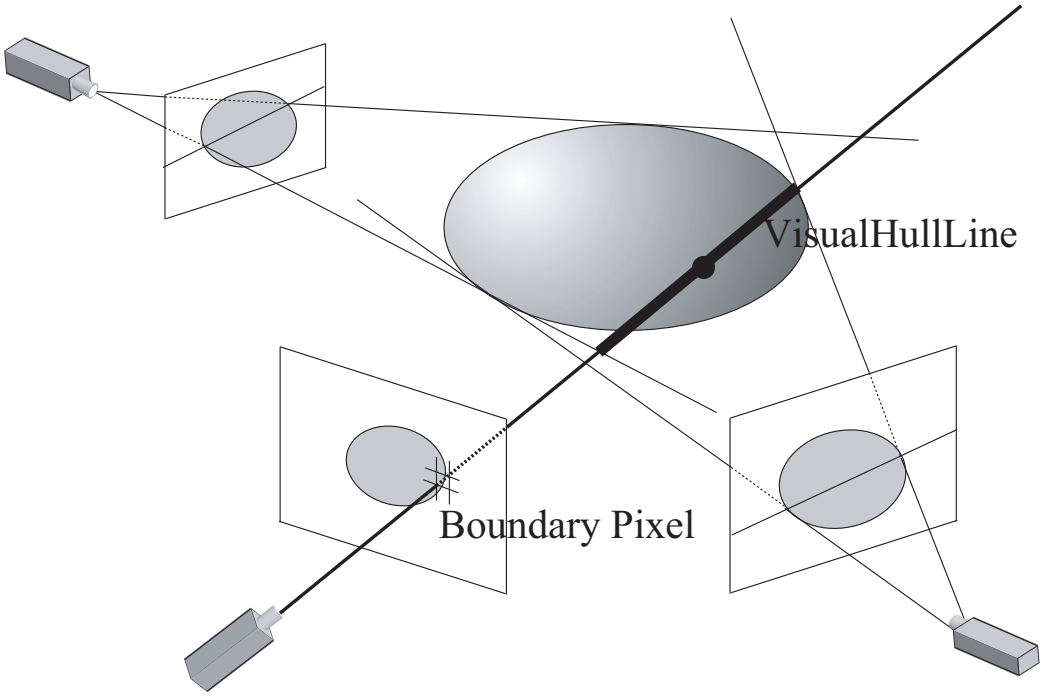


Figure 4.4: Visual Hull Line

- At more than one point, the visual hull line is tangent to the object.

These constraints are called *visual hull line constraints*.

Not satisfying the first constraint violates the nature of visual hull. The nature is that the visual hull circumscribes the object.

Not satisfying the second constraint also violates another nature of visual hull. The nature is that the silhouettes of the object completely correspond with silhouettes of the object's visual hull. When a visual hull line is not tangent to the object, the edge pixel of the visual hull line is not included in the silhouette. Such edge pixel violates the nature.

These visual hull line constraints give the silhouette constraint.

The use of visual hull constraints

Not all the visual hull line gives the silhouette constraint to a distance map. Some visual hull lines are visible by a camera, and some are not. Visual hull lines occluded by the other lines are not visible and give no constraint to the distance map.

The silhouette constraint to a distance map depends on a visibility of the visual hull lines.

The use of Z-buffer gives the decision of the visibility. Projecting all the visual hull lines to the camera image, we obtain the Z-buffer. Comparing Z-value (depth) of Z-buffer and that of visual hull line gives the decision.

A visual hull line which is visible by camera A, shown in Figure4.5(a), gives both of two visual hull line constraints to the distance map of camera A. On the other hand, invisible visual hull line, shown in Figure4.5(c), gives no constraint. A visual hull line which is partially visible, shown in Figure4.5(b), only gives the first constraint. It does not give the second constraint, a constraint that the visual hull line is tangent to the object at more than one point. The object might be tangent to the visual hull line, and its tangent point might be occluded by the other visual hull line. The existence of such occluded tangent point makes the second constraint useless.

The visible or partially visible visual hull lines give the constraint to some pixels on the distance map. Such pixels are extracted with the following procedure: First, projecting the visual hull line to the distance map, we obtain the 2D line. The 2D line intersects a grid of the distance map. The intersection points of the 2D line and the grid, shown in Figure4.5, are extracted. Using the Z-buffer, visible intersection points are extracted

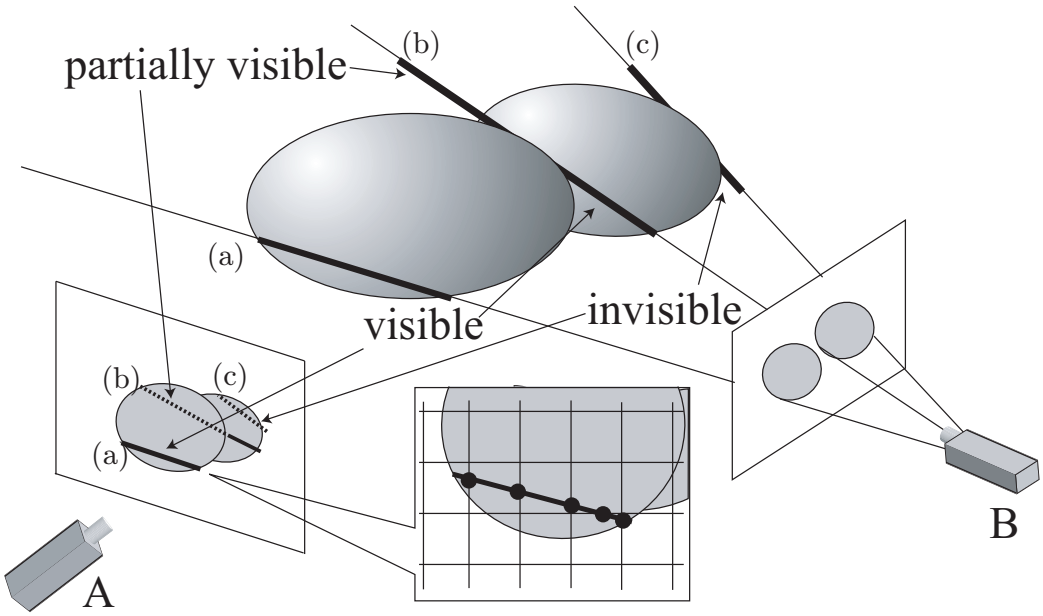


Figure 4.5: Visible/Invisible visual hull line

from the intersection points. We express the visible intersection points as $(v_{x1}, v_{y1}), \ldots, (v_{xK}, v_{yK})$. We evaluate the degree of satisfaction of the visual hull line constraint, comparing a depth of the visible intersection point and that of the visual hull line.

A depth of the visible intersection point (v_{xk}, v_{yk}), denoted by $Z(v_{xk}, v_{yk})$, is given by

$$Z(v_{xk}, v_{yk}) = Z(\lfloor v_{xk} \rfloor, \lfloor v_{yk} \rfloor) \cdot \left(1 + \frac{v_{xk} - \lfloor v_{xk} \rfloor}{f_c} p_k + \frac{v_{xk} - \lfloor v_{xk} \rfloor}{f_c} q_k\right) \tag{4.7}$$

where, $Z(\lfloor v_{xk} \rfloor, \lfloor v_{yk} \rfloor)$ is a depth of the pixel whose representative point is $(\lfloor v_{xk} \rfloor, \lfloor v_{yk} \rfloor)$, p_k and q_k are the component of surface normal $\mathbf{n}_k = (p_k, q_k, 1)^{\mathrm{T}}$.

Under the first visual hull line constraint, $Z(v_{xk}, v_{yk})$ is always deeper than a depth of the visible visual hull line.

Under the second visual hull line constraint, more than one of the $Z(v_{xk}, v_{yk})$ correspond to a depth of the visual hull line unless the visual hull line is occluded.

Let δZ_k be a difference between $Z(v_{xk}, v_{yk})$ and a depth of the visual hull line, and let $\delta Z_{\min} = \min(\delta Z_k)$. We get $\delta Z_k \geq 0$ for all k, using the first visual hull line constraint. We get $\delta Z_{min} = 0$ when the visual hull line is visible, using the second constraint. These two constraints give the following energy which expresses the visual hull line consistency.

$$E_{VL} = \begin{cases} 0 & if \ \ \Delta Z_{\min} \geq 0 \text{and} \\ & \text{the visual hull line is partially invisible} \\ {\Delta Z_{\min}}^2 & otherwise \end{cases} \tag{4.8}$$

Calculating E_{VL} of each visual hull line, and summing them up, we get the energy of the silhouette consistency.

4.3.3 Minimization

We minimize the following E_{all} and acquire the distance map. The energy E_{all} consists of the energy function based on the needle map consistency (described in Section4.3.1), and that based on the silhouette consistency (described in Section4.3.2). Simulated annealing[35] is used for the minimization.

$$E_{all} = E_N + \lambda \sum E_{VL} \tag{4.9}$$

where λ is a penalty of visual hull line constraint. A sufficiently large λ provides a distance map which satisfies the needle map constraint within the visual hull line constraint.

Simply speaking, minimizing E_{all} works in the following manner. Depths of pixels on the depth edge are estimated, attaching high weight to the silhouette consistency. On the other hand, depths of the other pixels are estimated, attaching high weight to the needle map consistency.

4.3.4 Unifying the Distance Maps

Integrating the distance maps of each camera provides the object's shape . The point $\mathbf{X}_i^c$ which corresponds to the representative point (x_i^c, y_i^c) is

calculated by

$$\mathbf{X}_i^c = \begin{pmatrix} \frac{Z(x_i^c, y_i^c)}{f_c}(x_i - e_x^c) \\ \frac{Z(x_i^c, y_i^c)}{f_c}(y_i - e_y^c) \\ Z(x_i^c, y_i^c) \end{pmatrix} \tag{4.10}$$

where f_c is a focal length of camera c, and (e_x^c, e_y^c) is a lense center.

$\mathbf{X}_i^c$, written in Equation4.10, is descrived in camera c coordinates. It is written in world coordinates as

$$\widetilde{\mathbf{X}_i^c} = R_c \mathbf{X}_i^c + \mathbf{T_c}$$

where R_c is a rotation matrix which rotates camera coorinates' axes to world coordinates', and $\mathbf{T_c}$ is a translation vector which translated camera coorinates' origin to world coordinates'.

4.4 Experiments

In this section, we show the experimental results of our method and discuss the accuracy of the method. We conducted two experiments; experiments with a synthetic data whose shape is known, and a real data.

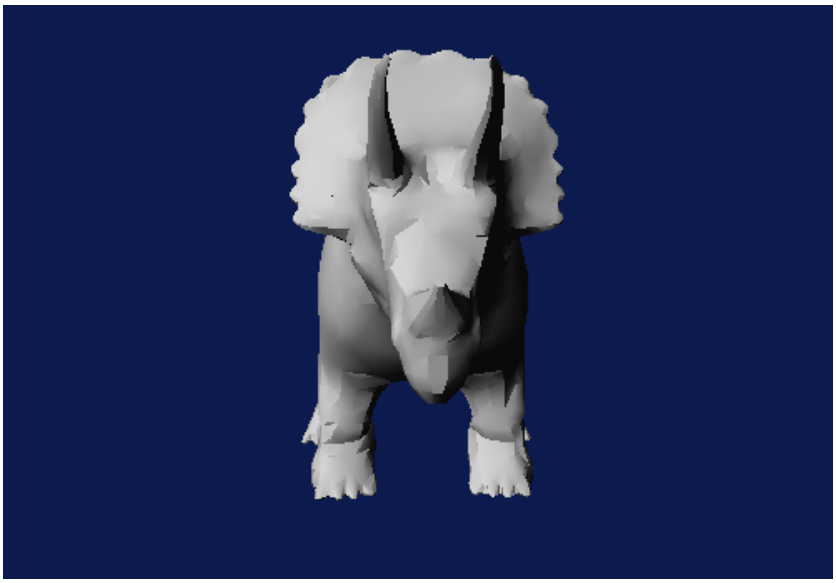
4.4.1 Experiments with Synthetic Data

A dinosaur model shown in Figure4.6(a)(b) is used for the experiment. Twenty directional lights were arranged and irradiated the model. Nine cameras which have 640×480 pixels were arranged around the model and observe the model.

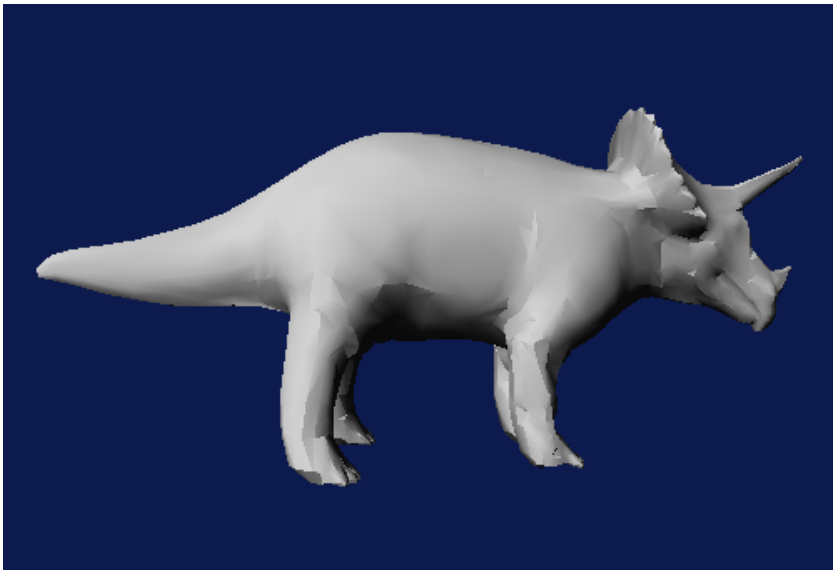
We first reconstructed the distance map of a camera (Figure4.7(c)). Then we reconstructed the shape of the model from the distance map. Figure 4.7(a), (b) and (c) show reconstructed shapes by minimizing three energy functions; consistency with silhouettes E_{VL}, that with needle map E_N, and that with E_{all} respectively.

Comparing Figure4.7(a) with (c) shows that needle map consistency produces smoother surface than silhouette consistency does.

A depth edge, which occurred on the border between left and right leg of the model, is observed on the image (Figure4.6(c)). The needle map consistency did not detect the depth edge and produced an unnatural shape

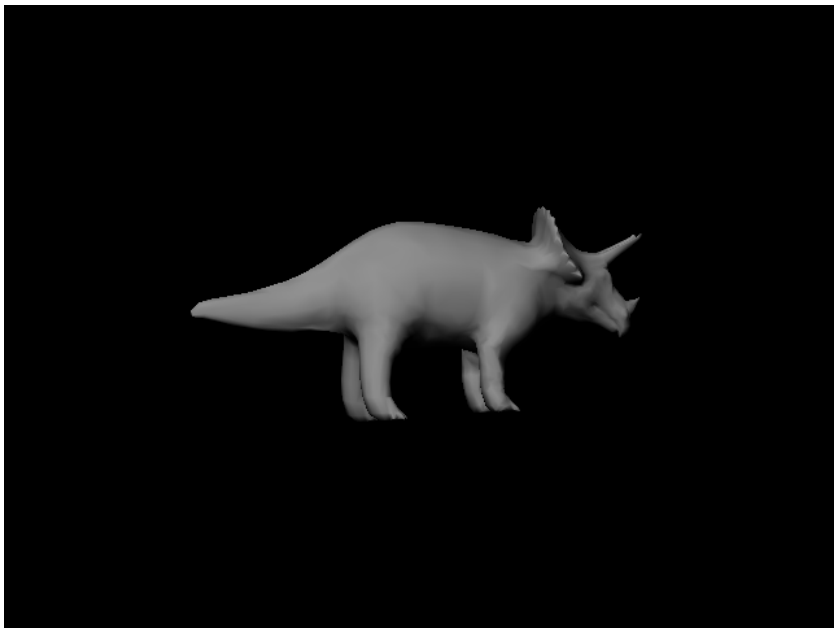


(a) object model(front)



(b) object model(side)

Figure 4.6: A synthetic data



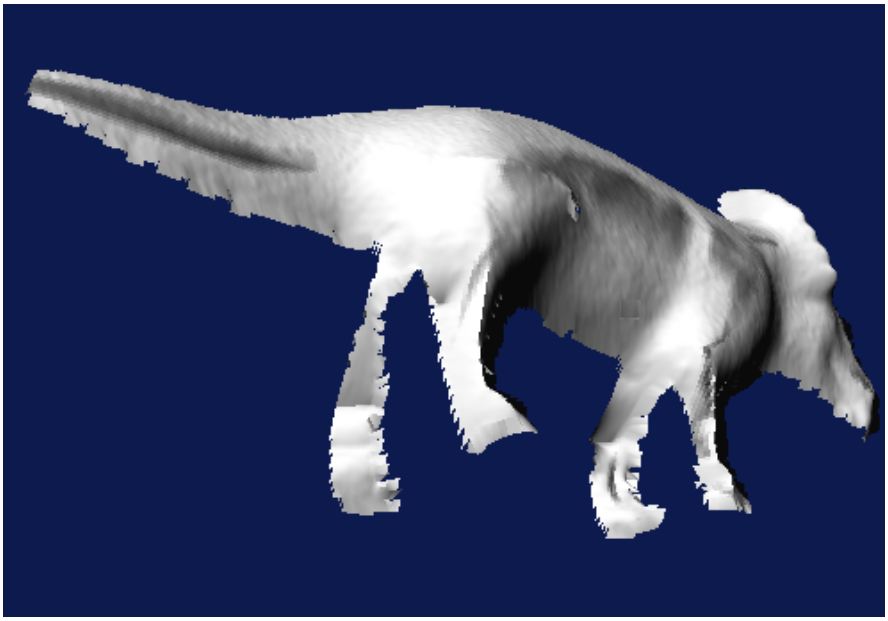
(c) input image

Figure 4.6: A synthetic data

(a) only silhouette consistency

(b) only needle map consistency

Figure 4.7: Partial shape from a depth map (synthetic data)



(c) proposed method

Figure 4.7: Partial shape from a depth map (synthetic data)

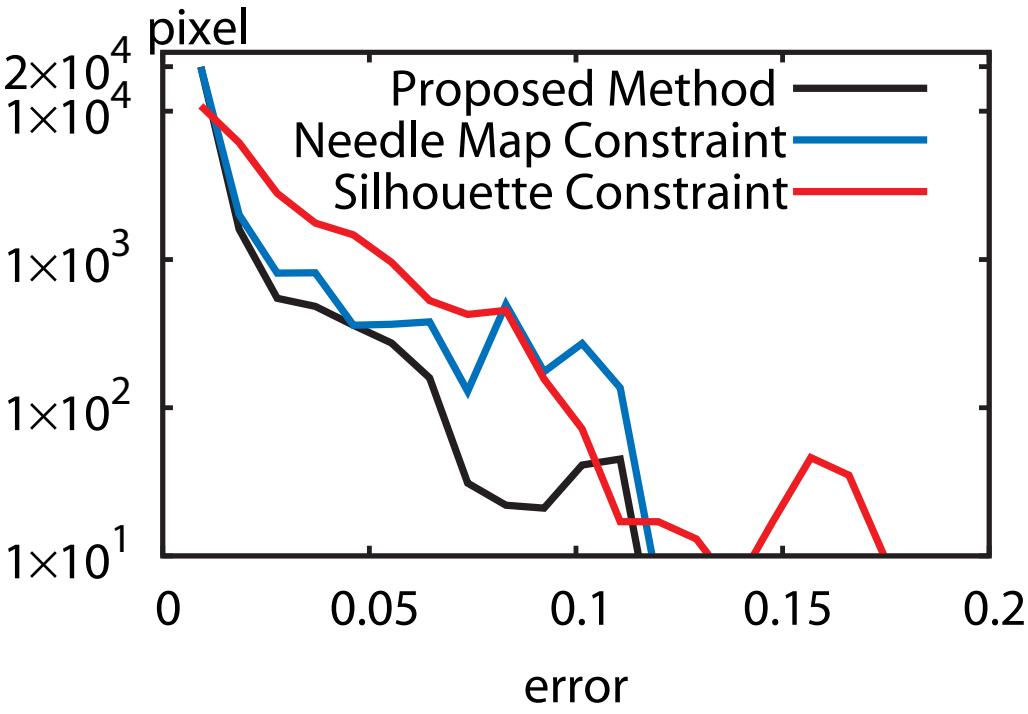


Figure 4.8: Error of depth estimation

shown in Figure4.7(b): The left and right legs are connected each other with smooth surface.

Figure4.7(c) shows that the silhouette consistency detected the depth edge and produced more natural shape than the needle map consistency did.

Errors in depth value estimation are shown in Figure4.8 and Table4.1. We calculate the error for each pixel m_i^c on the acquired depth image. The errors are normalized by the overall length of the model; error value 0.01 means that the difference between estimated depth and correct depth length is equal to 1% of the overall length of the model. Figure4.8 shows the log histogram of the error. Table4.1 shows the average and frequency distribution of depth error.

Using only the silhouette consistency results in that more than half of pixels contains error which is larger than 0.01. On the contrary, using the silhouette and needle map consistency reduces the number of such pixels to 15% of whole pixels. This reduction shows an effectiveness of using needle map consistency. Table4.1 shows that reconstructed depth image with our method has higher accuracy than that with needle map consistency has.

Figure4.8 shows that the number of pixels whose error is larger than 0.05 is reduced by our method. The large error is observed on the left legs on Figure4.7(b) and not observed on Figure4.7(c). So our method refines the unnatural shape and achieves higher accuracy of shape reconstruction.

Table 4.1: Depth Error.

	silhouette consistency	needle map consistency	our method
ave. error	0.019	0.011	0.0063
over 0.01	55.4%	21.5%	15.0%
over 0.02	31.9%	14.8%	8.4%
over 0.05	8.9%	7.0%	2.4%

4.4.2 Experiments with Real Data

An orange toy shown in Figure4.9 is used for the experiment. We put the toy into the center of our multi-camera system[27], and acquired input images with 8 cameras and 24 lights. Each camera has 640×480 pixels, and 4

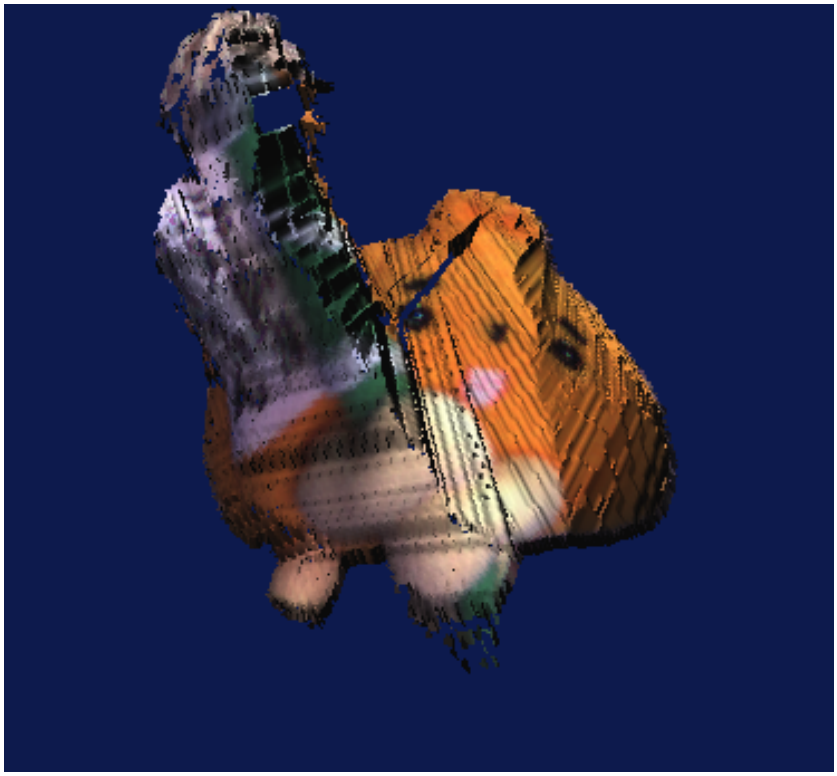
cameras are arranged on the front side of the toy and the other 4 camera are on the back side. Twelve lights illuminate the toy from the front side of the toy, and the other lights illuminate the toy from the back side.

We first reconstructed the distance map of a camera (Figure4.9). Then we reconstructed the shape of the model from the distance map. Figure 4.10 (a), (b), (c) show reconstructed shapes by minimizing three energy functions; consistency with silhouettes E_{VL}, that with needle map E_N, and that with E_{all} respectively.

Figure 4.9: input image

Comparing Figure4.10(a) with (c) shows the same fact that the results with synthetic data: Needle map consistency produces smoother surface than silhouette consistency does.

A depth edge, which occurred on the border between the body and pole of the toy, is observed on the image (Figure4.9). Comparing Figure4.10(b)



(a) only silhouette consistency

(b) only needle map consistency

Figure 4.10: Partial shape from a depth map(real data)

(c) proposed method

Figure 4.10: Partial shape from a depth map(real data)

with (c) shows that our method avoids bad effects of the depth edge on the shape reconstruction.

Figure4.11 shows reconstructed whole shape by integrating all cameras' depth map. As Figure4.11(a) shows, visual hull, which uses only the silhouette consistency, does not reconstructs smooth surface. Figure4.11(b) is a result by using needle map consistency and not using silhouette consistency. Figure4.11(b) shows that lack of silhouette consistency produces unnatural shape. On the contrary, using the silhouette and needle map consistency does not produce such unnatural shape, as Figure4.11(c) shows.

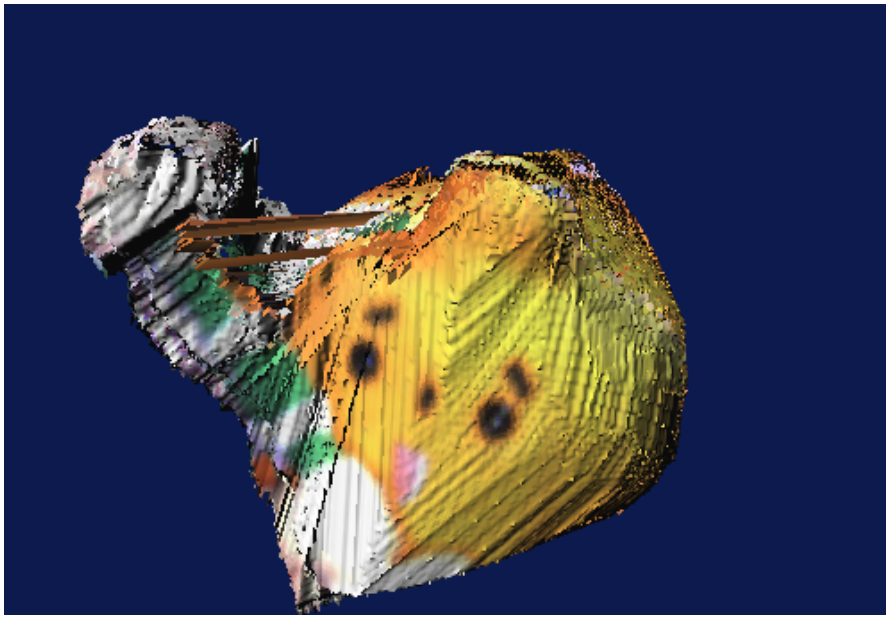
These results show the effectiveness of our method for objects on which depth edges exist.

4.5 Conclusion

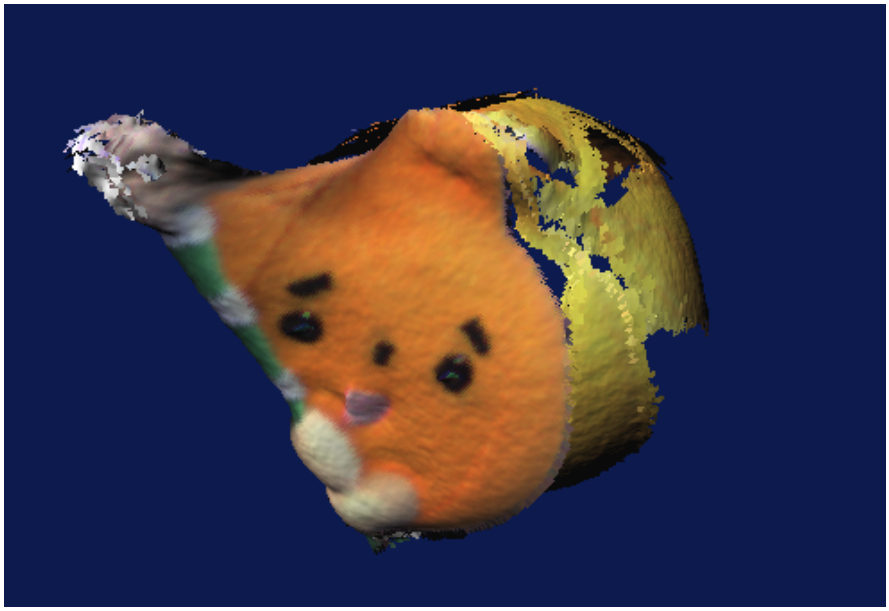
A shape reconstruction method which uses silhouettes and object's shading is proposed in this chapter. Our method can acquire a concave and a smooth surface, which the volume intersection method can not acquire.

Our method reconstructs depth maps from needle maps which are obtained with photometric stereo. The depth edges in the needle maps cause an incorrect reconstruction of the depth maps. We detect the depth edges by using the consistency between the depth maps and the silhouettes, and reduce the bad effect from the depth edges.

Experimental results show that a reconstructed shape with our method has small gaps observed between two depth maps. Coping with the gaps is one of the future works.

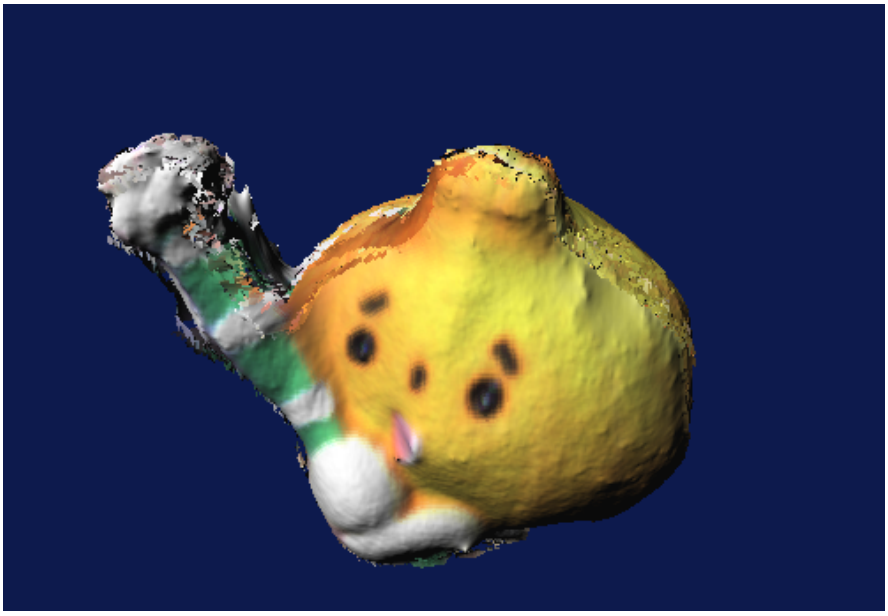


(a) visual hull



(b) only needle map constraint

Figure 4.11: Recovered shape



(c) proposed method

Figure 4.11: Recovered shape

Chapter 5

Conclusion

In this thesis, we proposed an approach for acquiring a 3D object model from silhouettes. The 3D object model is widely used in various applications. We categorized them according to the computation/observation time. There is a relationship between the computation/observation time and controllability of lighting environment. The first contribution of this thesis is to make the relationship clear.

Silhouette-based acquisition of 3D object models is hard task, for it requires restriction as to an object's shape or prepared object's shape. The second contribution of this thesis is to cope with the difficulty. The 3D object model consists of three properties, reflection properties, pose, and shape of the object. We proposed the approaches which acquires these properties.

First, a method for acquiring both the shape and the reflection properties under unknown lighting environment was proposed. The method does not require any prepared object's shape, and can be done by real-time processing. Previously proposed methods which acquire both the shape and the reflection properties required huge computational cost. Our method, on the contrary, is a voxel-independent method and reduces the computational cost. The use of the volume intersection method and simplifies Torrance-Sparrow reflection model enabled the voxel-independent calculation. Our method shows that real-time processing sacrifices accuracy of 3D object model; shape and reflection properties of convex surface can be acquired accurately, but those of concave surface can not be acquired.

Second, a method for acquiring both shapes of body parts and a motion of an articulated object was proposed. The method does not require any prepared shape of body parts. The unnecessary voxels and non-rigid

motion have bad effects on the acquisition. Our method employed a multi-dimensional voxel feature and probabilistic approach and reduced the bad effects. Our method shows that sacrificing a real-time processing enables us to acquire a motion of articulated objects in addition to the shape and reflection properties.

Finally, a method for acquiring the shape of objects was proposed. The volume intersection method had a restriction on the shape of objects; it could not acquire any concave shape. Our method does not require the restrictions, using the needle maps acquired by using photometric stereo. Depth edges have a bad effect on the shape acquisition, however. To reduce the bad effect, we used a consistency between the acquired depth maps and the silhouettes. This method shows that controllable lighting environment provides the shape of concave surfaces, sacrificing an adaptability of moving objects.

Several problems remain as future works.

Estimation of the Reflection Properties on Concave Surfaces Our method described in chapter2 used the volume intersection method to acquire the shape of objects. As we mentioned before, the volume intersection method can not acquire the concave surfaces. Estimation of the reflection properties on concave surfaces is one of our future works. The use of our method described in chapter4 may enable the estimation. However, it loses the voxel-independent calculation. Refinement of the algorithm of our method is required for the future work.

Acquisition of Non-rigid motion Acquisition of non-rigid motion is one of the future works. Our method described in chapter3 can produce an appearance with an arbitrary pose but produces the appearance around non-rigid region with low accuracy. The use of a meta-ball representation or superquadric function is a possible solution for the non-rigid motion acquisition.

Reduction of Gaps between Depth Maps Reduction of gaps between the depth maps is one of our future works. Experimental results shown in section4.4 show that a reconstructed shape with our method has small gaps, which are observed between two depth maps. Possible reasons of it are an error of camera's position, silhouette extraction, and surface normal

estimation. Several methods which cope with such gaps have already been proposed[8]. Applying these methods is our next future work.

Bibliography

[1] E. H. Adelson and J. R. Bergen. The Plenoptic Function and the Elements of Early Vision. In M. Landy and J. A. Movshon, editors, *Computation Models of Visual Processing*. MIT Press, Cambridge, 1991.

[2] A.Dempster, N.Laird, and D.Rubin. Maximum-likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B*, Vol. 39, No. 39, pp. 1–38, 1977.

[3] R. Basri and D Jacobs. Photometric stereo with general, unknown lighting. In *Proceedings of Computer Vision and Pattern Recognition*, Vol. 2, pp. 374–381, 2001.

[4] C.Y. Chen, R. Klette, and C.F. Chen. 3D Reconstruction Using Shape from Photometric Stereo and Contours. In *Proceedings on Proc. on IVCNZ*, pp. 251–255, 2003.

[5] Changsuk Cho and H.Minamitani. 3-D reconstruction using photometric stereo and silhouette informations. In *Proceedings of 20th International Conference on Industrial Electronics Control and Instrumentation (IECON '94)*, Vol. 2, pp. 806–809, 1994.

[6] Isaac Cohen and Mun Wai Lee. 3D Body Reconstruction for Immersive Interaction. In *Proceedings of Second International Workshop on Articulated Motion and Deformable Objects*, pp. 119–130, 2002.

[7] D.Anguelov, D.Koller, H.C.Pang, P.Srinivasan, and S.Thrun. Recovering Articulated Object Models from 3D Range Data. In *Proceedings of the Uncertainty in Artificial Intelligence Conference(UAI)*, pp. 18–26, 2004.

[8] Pau Gargallo and Peter F. Sturm. Bayesian 3D Modeling from Images Using Multiple Depth Maps. In *Proceedings of Computer Vision and Pattern Recognition*, Vol. 2, pp. 885–891, 2005.

[9] Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, pp. 721–741, 1984.

[10] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F.Cohen. The Lumigraph. In *SIGGRAPH 96 Conference Proceedings*, pp. 43–54, 1996.

[11] D.M. Gravila and L.S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 73–80, 1996.

[12] J.Gao, R.T.Collins, A.G.Hauptmann, and H.D.Wactlar. Articulated Motion Modeling for Activity Analysis. In *Proceedings of IEEE Workshop on Articulated and Nonrigid Motion*, p. 20, 2004.

[13] Ioannis A. Kakadiaris and Dimitris Metaxas. Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 18–20, 1996.

[14] Ioannis A. Kakadiaris and Dimitris Metaxas. 3D Human body model acquisition from multiple views. *International Journal of Computer Vision*, Vol. 30, No. 3, pp. 191–218, 1998.

[15] Yoshinari Kameda, Takeo Taoda, Koh Kakusho, and Michihiko Minoh. High Speed 3D Reconstruction by Pipeline Video Image Processing and Division of Spatio-Temporal Space. *IPSJ Journal*, Vol. 40, No. 1, pp. 13–22, 1999. (in Japanese).

[16] Takeo Kanade, Peter Rander, Sundar Vedula, and Hideo Saito. Virtualized Reality: Digitizing a 3D Time-Varying Event As Is and in Real Time. In Yuichi Ohta and Hideyuki Tamura, editors, *Mixed Reality, Merging Real and Virtual Worlds*, pp. 41–57. Springer-Verlag, 1999.

[17] K.Cheung, S.Baker, and T.Kanade. Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and

Motion Capture. In *Proceedings of Computer Vision and Pattern Recognition Conference*, pp. 77–84, 2003.

[18] S. Baker K.M. Cheung and T. Kanade. Visual Hull Alignment and Refinement Across Time: A 3D Reconstruction Algorithm Combining Shape-From-Silhouette with Stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 375–382, 2003.

[19] Kiriakos N. Kutulakos and Steven M. Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision*, Vol. 38, No. 3, pp. 199–218, 2000.

[20] Aldo Laurentini. How Far 3D Shapes Can Be Understood from 2D Silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2, pp. 188–195, 1995.

[21] Aldo Laurentini. How Many 2D Silhouettes Does It Tkae to Reconstruct a 3D Object ? *Computer Vision and Image Understanding*, Vol. 67, No. 1, pp. 81–87, 1997.

[22] Aldo Laurentini. Computing the visual hull of solids of revolution. *Pattern Recognition*, Vol. 32, pp. 377–388, 1999.

[23] M. Levoy and P. Hanrahan. Light Field Rendering. In *SIGGRAPH 96 Conference Proceedings*, pp. 31–42, 1996.

[24] Marc Levoy, Billy Chen, Vaibhav Vaish, Mark Horowitz, Ian McDowall, and Mark T. Bolas. Synthetic Aperture Confocal Imaging. *ACM Transactions on Graphics*, Vol. 23, No. 3, pp. 825–834, 2004.

[25] A. Maki, M. Watanabe, and C.S. Wiles. Geotensity: Combining motion and lighting for 3d surface reconstruction. *International Journal of Computer Vision*, Vol. 48, No. 2, pp. 75–90, 2002.

[26] W.N. Martin and J.K.Aggarwal. Volumetric Descriptions of Objects from Multiple Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, No. 2, pp. 150–158, 1983.

[27] IIYAMA Masaaki, KAMEDA Yoshinari, and MINOH Michihiko. 4pi Measurement System: A Complete Volume Reconstruction System for

Freely-moving Objects. In *Proceedings of IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2003)*, pp. 119–124, 2003.

[28] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-Based Visual Hulls. In *SIGGRAPH 2000 Proceedings*, pp. 369–374, 2000.

[29] L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. In *SIGGRAPH 95 Conference Proceedings*, pp. 39–46, 1995.

[30] Takeshi NAGASAKI, Toshio KAWASHIMA, and Yoshinao AOKI. Structure Estimation of an Articulated Object from Motion Image Analysis Based on Factorization Method. *The Transactions of the IEICE*, Vol. J81-D–II, No. 3, pp. 483–492, 1998. (In Japanese).

[31] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, pp. 611–634, 1991.

[32] M. Okutomi and T. Kanade. A Multiple-baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 4, pp. 353–363, 1993.

[33] Jihun Park, Sangho Park, and Jake K. Aggarwal. Human Motion Tracking by Combining View-Based and Model-Based Methods for Monocular Video Sequences. In *Proceedings of International Conference on Computational Science and Its Applications (ICCSA)*, pp. 650–659, 2003.

[34] Ramesh Raskar, Karhan Tan, Rogerio Feris, Jingyi Yu, and Matthew Turk. Non-Photorealistic Camera: Depth Edge Detection and Stylized Rendering Using Multi-Flash Imaging. *ACM Transactions on Graphics*, Vol. 23, No. 3, pp. 679–688, 2004.

[35] R.J.Woodham. Photometric Method for Determining Surface Orientation from Multiple Images. *Optical Engineering*, Vol. 19, No. 1, pp. 139–144, 1980.

[36] Kosuke Sato and Seiji Inokuchi. Three-Dimensional Surface Measurement by Space Encoding Range Imaging. *Journal of Robotic Systems*, Vol. 2, No. 1, pp. 27–39, 1985.

[37] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object Shape and Reflectance Modeling from Observation. In *SIGGRAPH 97 Conference Proceedings*, pp. 379–387, 1997.

[38] Steven M. Seitz and Charles R. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. In *Proceedings of Computer Vision and Pattern Recognition Conference*, pp. 1067–1073, 1997.

[39] S.A. Shafer. Using color to separate reflection components. *Color Research and Application*, Vol. 10, pp. 210–218, 1985.

[40] B. Stenger, P.R.S. Mendonca, and R. Cipolla. Model-based 3D tracking of an articulated hand. In *Proceedings of Computer Vision and Pattern Recognition*, Vol. 2, pp. 310–315, 2001.

[41] N. Vasconcelos and A.Lippman. Empirical Bayesian Motion Segmantation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 2, pp. 216–221, 2001.

[42] Nuno Vasconcelos and Andrew Lippman. Empirical Bayesian EM-based Motion Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 527–532, 1997.

[43] Lior Wolf and A. Shashua. On Projection Matrices P^k -¿ $P^2, k = 3, ..., 6$, and their Applications in Computer Vision. *International Journal on Computer Vision*, Vol. 48, No. 2, pp. 53–67, 2002.

[44] Yoshihiro Yasumuro, Qian Chen, and Kunihiro Chihara. Three-dimensional modeling of the human hand with motion constraints. *Image and Vision Computing*, Vol. 17, pp. 149–156, 1999.

[45] Y. Yu, P. E. Debevec, J. Malik, and T. Hawkins. Inverse Global Illumination: Recovering Reflectance Models of Real Scenes from Photograph. In *SIGGRAPH 99 Conference Proceedings*, pp. 215–224, 1999.

[46] Y. Yu and J. Malik. Recovering Photometric Properties Of Architectural Scenes From Photographs. In *SIGGRAPH 98 Conference Proceedings*, pp. 207–217, 1998.

[47] Z. Zhang. A Flexible New Technique for Camera Calibration. Technical Report MSR-TR-98-71, Microsoft Research, 1998.

List of Publications by the Author

Journal Articles

1. Masaaki Iiyama, Koh Kakusho, Michihiko Minoh, “Robust Depth Map Acquisition Against Depth Edges with Silhouettes Consistency,” Trans. of IEICE, Vol.J89-D No.7. (accepted)

2. Masaaki Iiyama, Koh Kakusho, Michihiko Minoh, “Articulate Object Model Acquisition from Visual Hull in Time Sequences,” Trans. of IEICE, Vol.J89-D No.6 .(accepted)

3. Masaaki Iiyama, Hirofumi Aoki, Yoshinari Kameda, Michihiko Minoh,”A Voxel-Independent Reconstruction Method of Object Shape and Its Reflection Property under Unknown Lighting Condition,” IPSJ Journal, Vol.42, No.12, pp.1385–3193, 2001.(in Japanese)

4. Masahiro TOYOURA, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, “An Accurate Shape Reconstruction Method by Integrating Visual Hulls,” Trans. of IEICE, Vol.J88-D-II, No.8, pp.1549–1563, 2005. (in Japanese)

5. Takuya FUNATOMI, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, “Accurate 3D Scanning of Trunk Swaying Human Body Parts,” Trans. of IEICE, Vol.J88-D-II, No.8, pp.1530–1538, 2005. (in Japanese)

6. S.Yamada, C.Uwabe, T.Nakatsu-Komatsu, Y.Minekura, M.Iwakura, T.Motoki, K.Nishimiya, M.Iiyama, K.Kakusho, M.Minoh, S.Mizuta,

T.Matsuda, Y.Matsuda, T.Haishi, K.Kose, S.Fujii, K.Shiota, "Graphic and movie illustrations of human prenatal development and their application to embryological education based on the human embryo specimens in the Kyoto collection," Developmental Dynamics, Published Online: 29 Nov, 2005.

Refereed Conference Presentations

1. Masaaki Iiyama, Yoshinari Kameda, Michihiko Minoh, "4π Measurement System: A Complete Volume Reconstruction System for Freely-moving Objects,", IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2003), p.119–124, 2003.

2. Masaaki Iiyama, Yoshinari Kameda, Michihiko Minoh, "Estimation of the Location of Joint Points of Human Body from Successive Volume Data", Proceedings on International Conference on Pattern Recognition (ICPR2000), pp.699-702, 2000.

3. Masahiro TOYOURA, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "Silhouette Refining for the Volume Intersection Method with Random Pattern Backgrounds," Meeting on Image Recognition and Understanding 2005 (MIRU2005), pp.1247-1254, 2005.

4. Masahiro TOYOURA, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "An Accurate Shape Reconstruction Method by Integrating Visual Hulls in Time Sequences," Meeting on Image Recognition and Understanding 2004 (MIRU2004), Vol.II, pp.139-144, 2004.

5. Takuya FUNATOMI, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "Accurate 3D scanning of trunk swaying human body", Meeting on Image Recognition and Understanding 2004 (MIRU 2004), pp.565–570, 2005.

Conference Presentations

1. Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "Shape from Silhouettes and Needle Maps," Techinical Report of IEICE PRMU, PRMU2005-271 P.81–86, 2006.

2. Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "3D Object Model Acquisition from Silhouettes," In Proceedings of 4th International Symposium on Computing and Media Studies, p.86–93, 2006.

3. Masahiro TOYOURA, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "Silhouette Refining for the Volume Intersection Method with Random Pattern Backgrounds," Proceedings of the 2005 IEICE General Conference, D-12-133, 2005.

4. Su CHEN, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "3D Shape Acquisition Based on Consistency between Object Region and Projections of the Visual Hull," Proceedings of the 2004 IEICE General Conference, 2004.

5. Takuya FUNATOMI, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "Analysis of trunk sway for human balancing by stereo measurement," Forum on Information Technology (FIT2004), No.3, K008, pp.407–408, 2004.

6. Masahiro TOYOURA, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "An Accurate Shape Reconstruction Method by Motion Tracking," The 9th the Virtual Reality Society of Japan Annual Conference, p.111-113. 2004.

7. Takuya FUNATOMI, Isao MORO, Masaaki IIYAMA, Michihiko MINOH, "Surface reconstruction from Point Cloud of Human Body by Using Clustering," Technical Report of IEICE MVE, MVE-2002-133 P.53-56, 2003.

8. M.M. Sein,M. Iiyama,M. Minoh, "Reconstructing the Arbitrary View of an Object Using the Multiple Camera System," 2003 International Symposium on Micromechatronics and Human Science, 2003.

9. Masahiro TOYOURA, Masaaki IIYAMA, Koh KAKUSHO, Michihiko MINOH, "An Accurate Shape Reconstruction Method by Motion Tracking," The 31st IIEEJ General Conference, 2003.

10. Masaaki IIYAMA, Yoshinari KAMEDA, Michihiko MINOH, "4 π Measurement System : A Complete Volume Reconstruction System for Freely-moving Objects," Proceedings of the 65th National Convention of IPSJ, 4T7A-1, No.5, pp.411–414, 2003.

11. Takuya FUNATOMI, Masaaki IIYAMA, Shinobu MIZUTA, Michihiko MINOH, "Generation of Trajectory for Local Self-intersection Free 3D Metamorphosis between Patch Models," Forum on Information Technology (FIT2002), No.3, J42, pp.285–286, 2002.

12. Massaki IIYAMA, Yoshinari KAMEDA, Michihiko MINOH, "Acquisition of a Model of an Articulate Object from Multiple Video," Forum on Information Technology (FIT2002), No.3, I8, pp.15–16, 2002.

13. Shingo MATSUMURA, Shinobu MIZUTA, Masaaki IIYAMA, Michihiko MINOH, "Shoulder Shape Representation with Posture Changes using Measured Data Sets," Tech. Report of IEICE PRMU, PRMU 2001-238, P.41–46, 2002.

14. Masaaki Iiyama, Yoshinari Kameda, Michihiko Minoh, "Acquisition of a Model of an Articulate Object from Successive Volume Data," Proceedings of the 190th Technical Meeting of IIEEJ, Vol.01-05, pp.43–48, 2001.

15. Masaaki IIYAMA, Hirohumi AOKI, Yoshinari KAMEDA, Michihiko MINOH, "A Parallel Computable Reconstruction Method of Reflection Property under Unknown Lighting Condition," Technical Report of IEICE PRMU, Vol.100, No.360, PRU2000-98, pp.17–22, 2000.

16. IIYAMA Masaaki, KAMEDA Yoshinari, MINOH Michihiko, "Estimation of the Location of Joint Points of Articulate Object from Successive Volume Data," Proceedings of the 2000 Information and Systems Society Conference of IEICE, D-12-71, pp.258, 2000.

17. IIYAMA Masaaki, KAMEDA Yoshinari, MINOH Michihiko, "Pose Estimation of Human Body on Voxel Data Considering Interference with Movable Area of Human Model Parts," Proceedings of the 1998 Information and Systems Society Conference of IEICE, D-12-85, pp.307, 1998.

Technical Magazine Articles

1. Masaaki IIYAMA, Michihiko MINOH, "3D Human Model Cetnered Framework,", Image Labo., No.2004-11, p.14–19, 2004.